Джина Гайд

МЕТОДОЛОГИЯ ВНЕДРЕНИЯ ИИ НА ПРЕДПРИЯТИИ

Внедрение ИИ в разработку ПО

Сокращение сроков поставки и снижение численности команд.

При сохранении требуемого качества
и удержании рисков в допустимых пределах.



Трудозатраты зависят от численности команд и сроков поставки.
Схема показывает направление изменений, а не прогноз результата.

Краткое руководство и рабочие материалы

Люди · Процессы · Технологии

Содержание

Материалы методологии собраны в одном документе. Нажми на название раздела или материала, чтобы перейти к нему. Архивные материалы не включены; ссылки на них открывают веб-версию.

Краткое руководство 4

- Внедрение ИИ в разработку ПО 4

Гайды процессов 31

- Управление целями и приоритетами внедрения 31
- Методология и передача опыта 33
- Управление составом и волнами внедрения 35
- Управление программой и ресурсами 37
- Сведение результатов и общие улучшения 39
- Общая инфраструктура и условия доступа 41
- Встраивание агентов в процессы разработки 43
- Управление внедрением в продуктовую команду 46
- Управление эффективностью работы команды и агентов 49
- Подготовка и вовлечение участников 57
- Обеспечение технической готовности 60

Гайды ролей 63

- Гайд роли: Спонсор 63
- Гайд руководителя программы 66
- Гайд методолога офиса внедрения 69
- Гайд роли: Владелец продукта 72
- Гайд роли: РП 79
- Гайд ИИ-инженера 91
- Гайд роли: Аналитик 113
- Гайд роли: Архитектор 119
- Гайд роли: Разработчик 123
- Гайд роли: Тестировщик 131
- Гайд роли: DevOps 139

Практические материалы · Howtos 146

Практические материалы 146
Офис внедрения ИИ 147
Словарь методологии 150
Агент и репозиторий внедрения 153
Показатели и правила расчёта 156
Автоматический сбор сведений и расчёт метрик 177
Проектирование тестовой инфраструктуры 181
Стратегия тестирования при внедрении ИИ в разработку ПО 206

Шаблоны и чек-листы 213

Паспорт программы внедрения 213
Паспорт программы «Агентская разработка в Меридиан Сервис» 216
Правила работы в Git 220
Правила Git: кабинет «Меридиан Сервис» 223
Карточка инфраструктуры 227
Инфраструктура внедрения в личный кабинет Меридиан Сервис 233
План мониторинга внедрения в продукт 246
План мониторинга личного кабинета Меридиан Сервис 251
Паспорт проекта внедрения в продукт 257
Паспорт проекта «Агентская разработка в личном кабинете Меридиан Сервис» 260
Жизненный цикл задач 265
Жизненный цикл задач: кабинет «Меридиан Сервис» 269
Чек-лист готовности к PoC 273
Чек-лист готовности к Пилоту 282
Чек-лист завершения Пилота 292
Чек-лист развития условий команды для работы с агентами 304
Условия команды кабинета «Меридиан Сервис» 308
Журнал улучшений агентов, процессов, инструментов и тестов 309
Улучшения при внедрении в кабинет «Меридиан Сервис» 311
Ежедневный журнал решения задач 312
Решение задач кабинета «Меридиан Сервис» за день 314
Журнал показателей внедрения 315
Показатели внедрения в кабинет «Меридиан Сервис» 317

01 Краткое руководство

Внедрение ИИ в разработку ПО

Методология описывает разработку ПО с использованием ИИ-агентов — **агентскую разработку** — на уровне предприятия и продуктовых команд. Далее используется краткое название «агент»; определения собраны в словаре.

Общее описание

Цель

ИИ в разработке ПО используется для целей **сокращения сроков поставки и снижения численности команд при производстве ПО** при сохранении требуемого качества и удержании рисков в допустимых пределах.

Это типовая обобщённая цель. При внедрении её уточняют с учётом продукта, условий и приоритетов предприятия. Метрики результата показывают прогресс достижения цели; метрики процесса и баланса помогают управлять внедрением: находить ограничения, выбирать изменения и замечать нежелательные последствия. поэтапное формирование целей и метрик · Определения и расчёты метрик.

Подход

Внедрение начинается с отдельных участков производства ПО и ограниченного набора типов задач. Выбирают работу с понятным результатом и способом проверки, готовят условия и подтверждают, что агенты справляются с ней в реальном продукте. Затем подключают соседние работы и более сложные задачи, связывая освоенные участки в полный производственный цикл. Например, от реализации небольшой правки по готовым требованиям переходят к её тестированию, обработке замечаний и подготовке результата к интеграции.

Расширение идёт одновременно в двух направлениях: внутри продукта растёт состав работ, выполняемых агентами, а на предприятии — число команд и продуктов, использующих проверенные практики. Следующий шаг выбирают по результатам предыдущего: качеству, срокам, нагрузке людей и выявленным ограничениям. Охват работ и степень человеческого контроля меняются раздельно: передача нового вида задач не означает автоматической отмены проверок.

Агенты автономно выполняют согласованные работы на общей, централизованно управляемой платформе и встраиваются в существующие производственные процессы и инструменты. Их навыки развиваются за счёт самообучения и работы ИИ-инженера. По мере накопления опыта уточняют инструкции, контекст, проверки и технические средства; подтверждённые улучшения позволяют расширять применение и автономность.

Залог успеха — **заряженная команда, ритмичная работа над улучшениями и технологии, проверенные практикой**. Подход PLAN → DO → CHECK → ACT применяется в нескольких связанных циклах. После результата задачи или MR разбирают затруднения: понятные проблемы

исправляют сразу, а неочевидные наблюдения сохраняют для совместного анализа. Раз в неделю накопленные факты группируют по причинам, выбирают наиболее полезные изменения и проверяют их эффект на следующих задачах.

Отдельно работают над временем и расходом ресурсов агентов, трудозатратами людей на HITL и ожиданием проверок. Цель — существенно уменьшить человеческий труд на сопоставимый объём поставленного ПО, сохраняя качество. Для этого перед изменением определяют ожидаемый эффект, после проверки — закрепляют улучшение, пересматривают его или отменяют. Не требуется устранить все обращения к людям: необходимый HITL сохраняют как часть рабочего процесса, а Пилот завершают по согласованным показателям и обязательным условиям. Циклы PLAN → DO → CHECK → ACT.

Люди, процессы, технологии

Внедрение рассматривается на двух уровнях: предприятия и продуктовой команды. На уровне предприятия проверяют целесообразность изменений, организуют общие условия и последовательное подключение команд. В каждой продуктовой команде готовят рабочий контур, проверяют агентов на задачах и переходят к их применению в текущей работе. Ниже сначала описаны этапы каждого уровня, затем необходимые изменения в людях, процессах и технологиях.

Предприятие продолжает производить ПО, а внедрение агентов меняет способ этого производства. Для каждого продукта определяют, что сохраняется, что добавляется и изменяется, а от каких работ и операций можно отказаться по мере появления проверенной замены.

Внедрение на уровне предприятия

Прогнозирование эффектов → Пилотная волна → Волны масштабирования → Подведение итогов.

Этап	Основная работа и результат
Прогнозирование эффектов	На ограниченных реальных задачах проверяют применимость агентов, затраты человеческого труда и технические ограничения. Готовят прогноз изменений производства ПО, затрат и рисков; по результатам принимают решение о запуске программы. Для первых проб полноценная программа и отдельный офис не требуются.
Пилотная волна	Фиксируют цели, ресурсы и состав первых продуктовых команд. В каждой проводят Подготовку, PoC и Пилот, обеспечивают общие условия работы и получают подтверждённые результаты для проверки прогноза.
Волны масштабирования	Проверяют переносимость опыта на другие продукты, подключают команды по готовности и возможностям сопровождения, развивают общие практики и инфраструктуру.
Подведение итогов	Сопоставляют достигнутый эффект, затраты и ограничения, принимают решения о продолжении и закрепляют порядок постоянной поддержки и развития. Итоги каждой волны уточняют следующую.

Выбор продуктов для пилота

Для успеха первого пилота крайне важно выбирать продукты типа «Разработка функциональности» с достаточным бэклогом минимум на несколько месяцев работы команды. Основной объём должны составлять задачи по развитию продукта: детализация фич, реализация, тестирование и исправление замечаний. Команда должна иметь устойчивую загрузку такими задачами; нехватка работы и ожидание внешних решений не позволят проверить эффект автономной разработки.

Продукты оценивают по характеру предстоящего потока работ. Один продукт в разные периоды может относиться к разным типам.

Тип работ	Что составляет основной объём	Пригодность для первого пилота
Разработка функциональности	Детализация фич, изменения кода, тестирование и исправления. Большинство необходимых решений принимается внутри команды.	Приоритетный выбор: можно последовательно передавать агентам связанные участки разработки и проверять результат.
Разработка с межкомандными зависимостями	Существенная разработка, для которой нужны решения, изменения или поставки других команд.	Подходит, если заранее обеспечены зависимости и выделен участок, который можно выполнять и проверять самостоятельно.
Организационное внедрение и настройка	Обследования, переговоры с подразделениями и смежными системами, согласование процессов и совместных планов, затем настройка готового решения. Собственной разработки немного.	Основную работу ведут люди; потенциал именно автономной разработки ограничен. Для первого пилота такой продукт не выбирают, если в нём нельзя выделить достаточный самостоятельный объём разработки.
Поиск решения	Выяснение потребностей и проверка предположений о том, какую функциональность создавать и какое поведение будет полезным.	Сначала проводят исследования и проверяют прототипы; для пилота выделяют сформировавшийся поток разработки.

Достаточность бэклога проверяют по перечню запланированных фич и объёму предстоящей разработки при текущем темпе команды. Не требуется заранее детализировать все постановки: уточнение требований вместе с агентами входит в работу. Важно, чтобы необходимые знания и решения были доступны внутри команды, а разработка не сводилась к нескольким небольшим изменениям после длительных согласований. При выборе учитывают и ожидаемое ускорение: в ходе пилота должен сохраняться достаточный поток задач для проверки результата.

Сроки задач должны позволять пройти несколько циклов проверки и исправления. Для пилота выбирают продукты, в которых есть достаточный поток задач с запасом времени до обязательной поставки. В планировании учитывают не только выполнение задачи агентом, но и несколько обращений к людям: уточнение требований, ревью, проверку тестов и повторную проверку исправлений.

Замечания возвращают агенту, чтобы он мог уточнить понимание задачи, получить недостающий контекст и самостоятельно довести решение до проверенного результата. Повторяющиеся ошибки и затруднения используют для улучшения инструкций, навыков и контекста следующих задач.

Срочные задачи, для которых неприемлемы повторные проверки и доработки, не должны составлять основной поток пилота. Если срок вынуждает передать задачу человеку, такую передачу фиксируют отдельно: она показывает ограничение текущего способа работы, но не подтверждает

неспособность агента завершить задачу после обратной связи.

Задачи берут в работу заблаговременно и выделяют время проверяющих. Запас до поставки нужен для исправлений, а ответы в HITL по-прежнему должны поступать быстро. Обязательное число попыток не задаётся: задача может быть решена сразу. Возможность повторных циклов сохраняется вместе с правом команды забрать задачу при необходимости.

Постепенность на предприятии — это переход от ограниченных проб к нескольким пилотным командам, а затем к волнам внедрения в других продуктах. Первые команды выбирают так, чтобы получить полезный результат и проверить существенные условия: технологии, зрелость процессов, требования к качеству и доступам. Удачный результат в одном продукте сначала проверяют в отличающихся условиях; после этого уточняют состав и ресурс следующей волны.

Команды могут находиться на разных этапах одновременно: одни проводят PoC, другие работают на Пилоте, третьи расширяют освоенный поток. Каждая проходит собственную проверку готовности, используя опыт предыдущих внедрений. Темп подключений определяется результатами и возможностями инфраструктуры и сопровождения. Если общие ограничения мешают работе, следующую волну уменьшают или откладывают до их устранения.

Охват внедрения и производственный эффект оцениваются раздельно: прохождение командами PoC и Пилота показывает продвижение внедрения, а сроки поставки и трудозатраты — его эффект. Результаты разных продуктов сводятся с сохранением различий условий.

Внедрение в продуктовые команды

Подготовка → PoC → Пилот → Масштабирование.

Этап	Основная работа и результат
Подготовка	Определяют участников и цели; собирают сведения об инфраструктуре, инференсе, процессах задач, Git, CI/CD и технических политиках. Готовят инфраструктуру и доступы, затем выбирают задачи PoC в целевом трекере и проверяют их чтение агентами.
PoC	ИИ-инженер ведёт взаимодействие с агентами, адаптирует их к продукту и проверяет решение задач. Специалистов продукта постепенно подключают через инженера. Ориентир: неделя 1 — онбординг; неделя 2 — простые задачи; недели 3–4 — типичные задачи будущего Пилота. Подтверждаются не только выполнение операций, но и качество результата.
Пилот	Производственная команда непосредственно ведёт текущий согласованный поток задач с агентами. ИИ-инженер помогает при затруднениях и развивает агентов. Ежедневно собираются сведения, еженедельно рассматриваются метрики, развиваются навыки и устраняются ограничения.
Масштабирование	Расширяют охват работ в продукте и проверяют новые режимы автономности. Агенты совершенствуются в режиме самообучения и за счёт целевой доработки навыков; опыт передаётся другим командам через процессы предприятия.

На PoC взаимодействие с агентами ведёт ИИ-инженер; на Пилоте — производственная команда. На PoC специалисты продукта предоставляют необходимые знания и решения, но ещё не принимают на себя регулярную работу с агентами.

ИИ-инженер передаёт задачи агентам, предварительно проверяя, как они понимают требования, допущения и ожидаемый результат. Он рассматривает MR и результаты тестирования, формулирует замечания и проводит повторные циклы исправлений. По выявленным затруднениям дорабатывает инструкции, навыки и правила описания задач. Так до подключения производственной команды отрабатываются не только выполнение задач, но и качество взаимодействия.

На последних задачах PoC специалистов продукта подключают через ИИ-инженера. Перед обращением к аналитику, разработчику или тестировщику инженер проверяет материалы агента: достаточно ли контекста, сформулирован ли вопрос, приложены ли результаты проверок. После ответа специалиста он проверяет, что замечания конкретны и доступны агенту, а агент правильно их понял и продолжил работу. Предметные решения остаются за соответствующим специалистом.

Получаются два последовательных уровня проверки: сначала ИИ-инженер проверяет готовность материалов к предметному рассмотрению, затем специалист продукта рассматривает их по существу. Инженер сопровождает обмен в обе стороны, чтобы не передавать коллегам неподготовленные результаты, а агентам — неоднозначные замечания.

К Пилоту команда должна быть обучена новому процессу. До участников доводят порядок работы и их действия, а ИИ-инженер демонстрирует постановку и уточнение задачи, ревью, возврат замечаний и повторную проверку исправлений на примере задач продукта. Сохраняют материалы обучения, состав участников и ссылку на демонстрацию либо её протокол. Самостоятельная работа с агентами и обязательная практическая проба на PoC не требуются; подключение специалистов через инженера остаётся возможностью.

На Пилоте регулярное взаимодействие передаётся производственной команде. Аналитики, разработчики и тестировщики непосредственно ставят и уточняют задачи, проверяют результаты и возвращают замечания. ИИ-инженер перестаёт быть обязательным посредником для каждого обращения, сохраняя помощь в сложных случаях и ответственность за развитие агентов.

Внутри продукта постепенно расширяют и набор типов задач, и состав работ агента. Например, сначала поручают небольшие изменения форм по готовым требованиям, оставляя человеку ревью и проверку поведения. Затем агент осваивает подготовку тестов, выполнение проверок и исправление замечаний. Следующий шаг — связанный цикл от уточнения и декомпозиции подготовленного запроса до проверенного изменения и разрешённой интеграции. Новые области, например внешние интеграции, проверяют отдельно.

На Подготовке выбирают исходный участок и задачи для PoC. На PoC проверяют этот маршрут, дорабатывают навыки агентов и определяют область Пилота. На Пилоте команда использует его в текущей работе и по результатам выбирает следующее расширение. Для каждого шага заранее определяют требования к контексту, навыкам, средам и проверкам, а после выполнения подтверждают качество и эффект. Ослабление человеческого контроля рассматривают отдельно: можно расширить состав работ с сохранением HITL или перевести уже освоенные работы на HOTL, не добавляя новых типов задач.

Переходы к PoC и Пилоту подтверждаются гейтами готовности. Успешное завершение Пилота фиксируется при достижении согласованных показателей и выполнении обязательных условий; оно не отменяет отдельного решения об изменении контроля или расширении внедрения. Чек-листы переходов.

Для выбранных работ различают **HITL** — человек проверяет или принимает решение до зависимого перехода; **HOTL** — асинхронный контроль без остановки движения; и работу без плановой ручной проверки. Типы работ и режим контроля меняются независимо. Новая задача, команда или модель не наследуют более высокую автономность автоматически. Термины контроля.

У агентов нет собственных настроенных ноутбуков: для работы нужны воспроизводимые среды, пригодные минимум для сквозной E2E-проверки выбранных сценариев и похожие на production по существенным свойствам. Пробелы в готовности команды развивают параллельно внедрению; необходимые условия конкретной задачи обеспечивают до её выполнения. Тестовая инфраструктура · Готовность команды.

Люди

Внедрение строится вокруг **автономных агентов, общих для продуктовой команды**. По способу организации работы агент похож на ещё одного участника команды: ему поручают задачу в рамках единого для агентов и людей процесса, он самостоятельно ведёт её к результату, задаёт вопросы, передаёт работу на проверку и исправляет замечания. Человеку не требуется отдельно запускать каждое действие и постоянно вести агента по шагам.

Знания и развитие агентов организуются централизованно. Требования к работе, решения по продукту и накопленный опыт сохраняются в общих инструкциях, навыках и контексте. Улучшение, сделанное по результатам одной задачи, должно помогать агентам при решении следующих задач всей команды.

Конфигурация внедрения

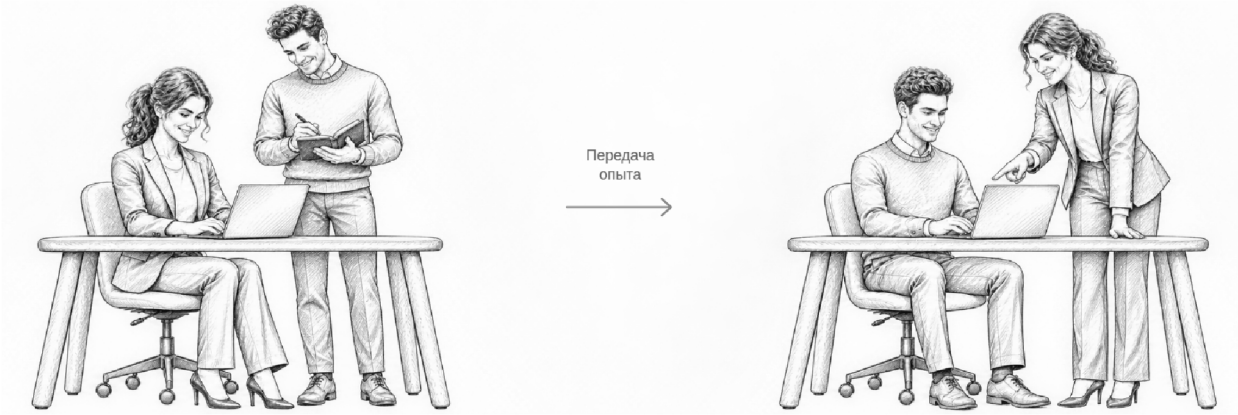
Самостоятельное внедрение. Продуктовая команда организует внедрение своими силами: назначает исполнителей ролей, осваивает платформу и методологию, настраивает агентов и развивает необходимые компетенции. На эти работы выделяется время наряду с текущими задачами продукта.

Внедрение с выделенной командой. К продуктовой команде подключается команда внедрения. Это может быть внутренняя команда предприятия, прошедшая обучение, либо привлечённая профессиональная команда с опытом внедрений, глубоким знанием платформы и значительной практикой разработки ПО. Ей поручают согласованные работы: организацию внедрения, развёртывание платформы, настройку и адаптацию агентов, развитие средств тестирования и обучение специалистов продукта.

Почему выделенная команда помогает быстрее получить результат. Совместное внедрение обычно проходит быстрее и результативнее: сотрудники предприятия перенимают опыт на собственном проекте, решая реальные задачи вместе с опытными специалистами. Они осваивают настройку агентов, работу с требованиями и проверками, диагностику ошибок и улучшение навыков сразу в условиях своего продукта. Накопленные практики помогают сократить пробы и ошибки, а совместная работа готовит команду к дальнейшему сопровождению.

На этапе PoC опытная команда внедрения выполняет работы, а продуктовые команды набираются опыта.

На этапе «Пилот» продуктовые команды самостоятельно выполняют работы, а команда внедрения помогает со сложными задачами.



На PoC ИИ-инженер ведёт взаимодействие с агентами и показывает рабочие приёмы; на последних задачах специалисты продукта пробуют свои действия через инженера, который проверяет материалы и обратную связь. **На Пилоте** команда продукта ведёт рабочий поток, а специалисты внедрения помогают в сложных случаях и обучают коллег по ходу работы. Конкретные обязанности передаются по мере освоения; инженерное сопровождение может сохраняться.

В обеих конфигурациях специалисты продукта участвуют в постановке задач, предоставляют знания о продукте, проверяют результаты и принимают решения в своей зоне ответственности. Привлечение команды внедрения дополняет эту работу. Конфигурацию и состав участников фиксируют в паспорте проекта, общие условия и участников программы — в паспорте программы, конкретные работы и их исполнителей — в трекере на основе чек-листов этапов. По мере освоения работы распределение можно менять, сохраняя доступное сопровождение.

Команда

Роли описывают ответственность, а не отдельные штатные должности. На одного человека можно назначить несколько ролей; одну роль могут выполнять несколько участников с распределёнными работами.

Роли уровня предприятия

Роль	За что отвечает
Спонсор	Инициирует внедрение, утверждает цели и ресурсы, принимает решения о продолжении при отклонениях.
Руководитель программы	Координирует внедрения продуктовых команд, общие зависимости и ресурсы предприятия.
Методолог офиса внедрения	Задаёт общие требования к моделям, инфраструктуре и метрикам; обеспечивает соблюдение корпоративных правил, сопоставимость результатов и методологическую помощь.

Роли уровня продуктовой команды

Роль	За что отвечает
РП	Организует работы внедрения, обеспечивает участие людей, разбор препятствий и фиксацию результатов этапов.
Владелец продукта	Определяет продуктовые приоритеты, ожидаемое поведение и границы результата.
ИИ-инженер	Настраивает и развивает агентов, поддерживает инструменты и расчёт показателей внедрения.
Аналитик	Описывает и уточняет требования, связывает замысел с проверяемым поведением продукта.
Разработчик	Разрабатывает и проверяет код, обеспечивает технические правила и интеграцию изменений.
Архитектор	Определяет архитектурные ограничения и согласует технические решения в своей области.
Тестировщик	Определяет сценарии и достаточность проверок, развивает автотесты и проверяет результаты.
DevOps	Готовит и поддерживает платформу, инфраструктуру, конвейер разработки и тестовые среды.

На роли могут назначаться специалисты команды продукта или команды внедрения. Команда внедрения может быть внешней, внутренней либо смешанной. Спонсор, владелец продукта и руководитель программы назначаются со стороны предприятия с необходимыми полномочиями; специалисты внедрения помогают готовить материалы и выполнять работы, не подменяя их решения.

В разделе команды паспорта фиксируют роли, людей, принадлежность и выделенное время. Обязанности и полномочия следуют из описания роли. В проектном плане у каждой работы один конкретный ответственный и при необходимости соисполнители с указанным вкладом. Агент внедрения предлагает назначения по подсказкам работ; подтверждённое РП распределение сохраняется в плане. Совмещение ролей не требует передачи сведений самому себе: сохраняются документы, проверки и решения.

Изменения в работе специалистов

Какие изменения нужны:

- **Аналитик или другой автор замысла** обсуждает задачу с агентом и проверяет, как тот её понял. Для подробной постановки проверяет интерпретацию требований, допущения и открытые вопросы. Если работа начинается с замысла — проверяет и уточняет подготовленную агентом постановку до передачи в реализацию.
- **Разработчик** участвует в HITL при ревью MR: проверяет изменения агента, принимает их или возвращает с конкретными замечаниями. Выявленные проблемы используются для развития навыков разработки у агентов.
- **Тестировщик** участвует в HITL при проверке автотестов и результатов их прогонов: оценивает, что проверено, какие сценарии пропущены и достаточно ли оснований считать задачу выполненной. Выявленные пробелы используются для развития тестовых навыков агентов.

- **ИИ-инженер** — новая роль, отвечающая за развитие агентов. Разбирает все замечания, ошибки и затруднения, улучшает инструкции и навыки агентов, контекст и инструменты, затем проверяет, помогли ли изменения.
- **РП** обеспечивает быстрые ответы в точках HITL: назначает проверяющих, организует замещение и разбирает зависшие запросы. Также организует передачу знаний агентам: закрепляет ответственных за AGENTS.md и другие инструкции, обеспечивает доступ к решениям встреч и контролирует актуальность переданных сведений.

Подробности: участники внедрения, работа ИИ-инженера.

Процессы

Общий процесс позволяет постепенно передавать задачи автономным агентам. Люди и агенты работают в едином производственном потоке: агентам поручают выбранную часть задач, остальные выполняют специалисты команды. Сначала агентам передают ограниченный набор типов задач и участков работы. По мере освоения этих задач, развития навыков и подтверждения качества результатов команда расширяет область применения и увеличивает долю задач, выполняемых агентами.

Для людей и агентов сохраняются общие правила постановки, движения и приёмки задач. Изменения проходят через Git, MR и предусмотренные проверки. Агент получает работу через согласованный статус задачи или разрешение в чате и самостоятельно выполняет доступные ему шаги. Вопросы, результаты проверок и замечания связываются с той же задачей.

Автономность задаётся для конкретных работ и решений. Процесс определяет, что агент может сделать самостоятельно, где требуется блокирующее решение человека — HITL, а где применяется асинхронный контроль — HOTL. Запросы человеку поступают через чат с указанием события и ссылки на задачу или результат. Увеличение доли задач агентов и изменение режима человеческого контроля рассматриваются отдельно.

Какие изменения нужны:

- **Постановка задач.** Перед реализацией проверяются понимание требований и достаточность декомпозиции. Для выбранных типов задач определяются исходные сведения и проверяемый результат.
- **Движение работы.** В общем процессе задач и Git определяются точки HITL — события, при которых агент передаёт результат или вопрос человеку и ожидает его решения перед продолжением зависимой работы.
- **Проверки и решения.** Для каждой точки HITL или HOTL определяются предмет проверки и ответственная роль. Повторяющиеся проверки автоматизируются; изменение человеческого контроля принимается отдельно для выбранной области.
- **Улучшение работы.** Ежедневно собираются сведения о решении задач, участии людей и остановках. На недельном разборе команда рассматривает метрики результата, процесса и баланса и назначает изменения в агентах, правилах работы и тестовых средах.

Подробности: мониторинг и действия по результатам.

Технологии

Агентам предоставляется общая среда для самостоятельной работы. Платформа обеспечивает постоянную доступность агентов, выполнение задач, сохранение состояния и взаимодействие с людьми через чат. Агент получает необходимые знания о продукте и доступ к инструментам, может

собрать ПО, развернуть его в тестовой среде, выполнить проверки и продолжить работу по их результатам или после ответа человека.

Среда выполнения и проверки управляется централизованно. Версии инструкций и навыков, доступы, результаты запусков и изменения доступны ответственным участникам. Для каждой задачи можно установить, с какими условиями работал агент, что он сделал и на основании каких проверок получил результат.

Какие изменения нужны:

- **Платформа и инференс.** Обеспечить выполнение длительных задач, восстановление после остановок и достаточные мощности выбранных моделей. Настроить полномочия агентов, ограничения выполнения и возможность остановить работу.
- **Контекст и навыки.** Подключить актуальные требования, код, документацию и решения по продукту. Хранить инструкции и навыки с версиями, обеспечить их обновление и применение в следующих задачах.
- **Производственный конвейер.** Предоставить агентам программный доступ к трекеру, Git, CI/CD и другим необходимым системам. Обеспечить аккаунты, полномочия и сетевые подключения, согласованные с ИБ и пригодные для автономной работы.
- **Тестовые среды.** Подготовить воспроизводимые среды, пригодные минимум для E2E-проверок и близкие к production по существенным свойствам. Обеспечить программное развёртывание, сброс и подготовку данных, управляемые моки, эмуляторы оборудования и виртуальное время там, где это требуется для проверки продукта.

Подробности: ИИ-платформа, доступы и интеграции, тестовая инфраструктура.

В продуктовой команде эти изменения проверяются на реальных задачах и постепенно распространяются на новые участки работы. На уровне предприятия обеспечиваются общие платформа и инференс, ресурсы сред и условия доступа. При масштабировании офис внедрения обеспечивает применение корпоративных требований к моделям, инфраструктуре и метрикам, сопоставимость результатов и оптимизацию общих расходов; организует методологическую помощь, обучение и перенос практик между командами.

Гайды

По процессам

На уровне предприятия

Процессы предприятия обеспечивают общие условия внедрения и связывают результаты продуктовых команд с целями предприятия. На этапе прогнозирования эффектов достаточно ограниченных проб; полноценная программа и офис не являются условием их начала. Пилотная и последующие волны состоят из внедрений в командах, каждая из которых проходит Подготовку, PoC и Пилот. Итоги рассматриваются после каждой волны и при завершении программы.

Управление целями и приоритетами внедрения

Спонсор и руководитель программы совместно с владельцами продуктов определяют, какой эффект нужен предприятию, где он достижим и какие вложения оправданы. Прогноз уточняется по мере получения результатов команд.

Прогнозирование эффектов. Выбирают области, в которых сокращение сроков поставки и численности команд может дать значимый результат. Исследовательская группа из инициатора, ИИ-инженера и специалистов продукта проводит ограниченные пробы на реальных задачах, фиксирует помощь людей и ограничения. Уточняют исходное состояние, предполагаемый масштаб применения, необходимые изменения и затраты. Итог — проверенные предположения и решение, есть ли основания готовить программу; единичная удачная задача не доказывает эффект предприятия.

Пилотная волна. Прогноз переводят в цели первых внедрений: область работ, ожидаемое изменение показателей, условия качества и срок проверки. Отделяют экономию в производстве ПО от затрат на инфраструктуру, сопровождение и улучшения. Сопоставляют прогноз с результатами команд, выявляют условия, при которых эффект достигается. Итог — уточнённое ожидание для следующих внедрений и перечень необходимых вложений.

Волны масштабирования. Проверяют прогноз в других продуктах и при росте числа команд. Учитывают стоимость адаптации, нагрузку сопровождения и общие мощности. Для каждой волны определяют ожидаемый вклад и условия продолжения; если результат хуже ожидаемого, меняют область, ресурсы или темп расширения. Эффект не распространяют на неподтверждённые виды работ автоматически.

Подведение итогов. Сопоставляют достигнутые изменения с исходными целями и общими затратами. Отдельно показывают подтверждённый эффект, оставшийся потенциал и условия его получения. Спонсор решает, продолжать ли вложения, изменить приоритеты или завершить программу. Результат — обоснованное решение о дальнейшем применении, а не только отчёт о числе подключённых команд.

Подробности: гайд процесса, поэтапное формирование целей и метрик, этапы программы.

Управление составом и волнами внедрения

Руководитель программы с владельцами продуктов определяет, какие команды подключаются, в какой последовательности и при каких условиях. Процесс согласует охват с переносимостью опыта и возможностями сопровождения.

Прогнозирование эффектов. Выбирают продукты и задачи для первых проб: есть полезное применение, доступные специалисты и возможность проверить результат. Отмечают существенные различия будущих внедрений — технологии, зрелость процессов, требования к качеству и ограничения доступа. Итог — круг кандидатов и понимание, что предстоит проверить; полный список всех команд предприятия для начала не нужен.

Пилотная волна. Выбирают готовые к участию команды, которые представляют важные условия предприятия. Согласуют локальные цели, сроки, ресурс сопровождения и общие зависимости. Команды проходят собственные Подготовку, PoC и Пилот; начало работ не объявляется завершённым внедрением. К концу волны известны результаты, ограничения и комплект опыта, пригодный для переноса.

Волны масштабирования. Сначала проверяют переносимость практик в командах с существенными отличиями. Затем составляют очередную волну с учётом доступности ИИ-инженеров, инфраструктуры и специалистов продуктов. Новая команда проверяет собственную готовность, а режим автономности подтверждает отдельно. При накоплении препятствий уменьшают волну, меняют порядок или приостанавливают подключения. Результат — расширение, которое предприятие способно обеспечить.

Подведение итогов. Сверяют фактический состав и результаты с планом волны. Показывают доли команд, успешно перешедших от PoC к Пилоту и завершивших Пилот, причины остановок и условия дальнейшего подключения. Определяют следующие команды либо закрывают область программы. Итог — актуальный состав внедрений и решение о следующей волне; процент охвата не подменяет оценку производственного эффекта.

Подробности: гайд процесса, выбор первых команд, волны программы.

Управление программой и ресурсами

Спонсор обеспечивает полномочия и ресурс, руководитель программы координирует исполнение общих решений. При масштабировании офис внедрения поддерживает команды и организует совместную работу участников предприятия.

Прогнозирование эффектов. Инициатор собирает небольшую исследовательскую группу, определяет вопрос проверки, допустимые затраты времени и ресурсов, срок разбора и основания остановки. Используют доступные системы и существующие способы фиксации решений. Итог — проведённые пробы и рекомендация; паспорт программы, комитет и отдельный офис до решения о внедрении не требуются.

Пилотная волна. Назначают спонсора, руководителя программы и ответственных за общие направления. Согласуют бюджет, время ИИ-инженеров и участников продуктов, порядок рассмотрения зависимостей и принятия решений. Формируют общий план: локальные действия остаются у команд, межкомандные зависимости — на уровне программы. Первые пилоты могут работать без отдельного офиса, если функции координации и помощи обеспечены.

Волны масштабирования. Организуют постоянную функцию офиса с выделенным временем и полномочиями. Распределяют сопровождение, ведут очередь общих вопросов, планируют мощности и нагрузку специалистов. Рабочая сверка проводится еженедельно; комитет рассматривает существенные препятствия, ресурсы и расширение, например ежемесячно и перед новой волной. Решения содержат ответственного, срок и способ проверки исполнения. Недостаток ресурса изменяет план подключений.

Подведение итогов. Проверяют выполнение обязательств, фактические затраты и оставшиеся зависимости. Для постоянной работы назначают владельцев поддержки, инфраструктуры и дальнейшего развития, обеспечивают им ресурс. Незавершённые обязательства передают с ответственными и сроками. Результат — завершение программы или согласованный следующий период работы без потери управления и поддержки команд.

Подробности: гайд процесса, ответственность на предприятии, организация офиса, паспорт программы.

Общая инфраструктура и условия доступа

Владельцы платформы, инференса, инфраструктуры и систем конвейера вместе с ИБ обеспечивают технические условия автономной работы. Команды определяют потребности своего продукта, предприятие устраняет общие ограничения.

Прогнозирование эффектов. Проверяют, доступна ли конфигурация для первых проб: модели, выполнение агентов, код, данные и среда проверки. Определяют допустимые контуры и материалы. Не требуют заранее развернуть полную платформу предприятия; ограничения временной конфигурации сохраняют в выводах. Итог — условия ограниченной проверки и предварительный перечень инфраструктурных препятствий.

Пилотная волна. Выбирают общую схему платформы и инференса, обеспечивают мощности, поддержку и рабочие подключения первых команд. Согласуют аккаунты, сетевые пути и полномочия из контура агентов. Если правила допускают только персональные сеансы или запрещают необходимые связи, владельцы систем и ИБ определяют допустимый автоматизированный способ. Команды получают ресурс для тестовых сред, пригодных минимум для сквозных проверок. Итог — работающие общие сервисы и подтверждённые подключения пилотов.

Волны масштабирования. Планируют мощности и параллелизм по составу следующей волны, наблюдают доступность, ожидание инференса и стоимость. Обеспечивают воспроизводимое подключение команд, изоляцию проектов, выдачу и отзыв доступов, диагностику и обработку сбоев. Общие обновления проверяют до распространения; различия продуктов учитывают в их средах. Задержки ресурсов рассматривают как ограничения внедрения с ответственными и сроками устранения.

Подведение итогов. Проверяют достаточность и стоимость инфраструктуры для постоянной работы, фиксируют владельцев сервисов и порядок поддержки и обновлений. Неиспользуемые доступы и ресурсы пересматривают, оставшиеся ограничения включают в план развития. Результат — поддерживаемая инфраструктура с понятными полномочиями, ответственностью и условиями подключения следующих команд.

Подробности: гайд процесса, ИИ-платформа, конфигурация и доступы.

Методология и передача опыта

Офис внедрения вместе с ИИ-инженерами и представителями продуктов превращает опыт команд в применимые инструкции, навыки и способы работы. Процесс отвечает за содержание практик и обучение их применению.

Прогнозирование эффектов. Используют доступную методологию и материалы продукта для проведения проб. Сохраняют условия, полезные решения, ошибки и объём помощи людей. Отмечают, какие инструкции и знания понадобились и чего не хватило. Итог — исходные наблюдения для пилотной волны; опыт одной задачи не объявляется универсальной практикой.

Пилотная волна. Адаптируют материалы к продуктам и обучают участников на реальной работе. По итогам собирают комплект для переноса: инструкции по ролям, навыки, требования к средам и доступам, проверочные сценарии и известные ограничения. Отделяют общую часть от настроек конкретного продукта. Для материалов определяют владельцев и версии, чтобы следующая команда получила проверенный состав, а не случайный набор файлов.

Волны масштабирования. Перед применением комплекта сравнивают условия нового продукта с областью, в которой практика проверена. Адаптируют различающиеся части и подтверждают их работоспособность. Офис организует разбор случаев, обучение и обмен опытом; удачные изменения команд рассматривает для общего использования. Самообучение агентов и работа ИИ-инженеров пополняют знания, но перенос изменений в общий комплект требует проверки. Результат — накопление повторно применимых практик с явной областью действия.

Подведение итогов. Закрепляют актуальные версии методологии, навыков и учебных материалов, назначают ответственных за поддержку. Сохраняют причины принятых решений, неудачные подходы и условия применимости, выводят устаревшие версии из текущего использования. Определяют, как новые команды получают материалы и помощь. Итог — воспроизводимый способ передачи опыта, который сохраняется после завершения отдельных внедрений.

Подробности: гайд процесса, методологическая поддержка офиса.

Сведение результатов и общие улучшения

Офис внедрения собирает результаты команд и общих служб, выявляет ограничения предприятия и готовит предложения для решений. Руководитель программы обеспечивает выполнение общих изменений; локальные улучшения остаются в командах.

Прогнозирование эффектов. Определяют, какие наблюдения нужны для решения о программе: пригодность результата, помощь людей, затраты и ограничения первых проб. Фиксируют исходные условия и источники. Итог — понятная запись результатов и основания прогноза; полноценная регулярная отчётность предприятия на этом этапе не требуется.

Пилотная волна. Согласуют минимальный состав сводок команд: цели, область работ, этап, период, качество, затраты и ограничения данных. Проверяют единообразие определений и связь с первичными записями. Отделяют подтверждённые возможности PoC от эффекта Пилота, факт — от расчёта и ожидания. Несопоставимые продукты показывают отдельно. Итог — первая сводка, из которой видно, что достигнуто и какие общие условия мешают командам.

Волны масштабирования. Регулярно сводят продвижение по этапам, производственные результаты и нагрузку общих служб. Не усредняют проценты улучшения разных продуктов без основания и не учитывают общие затраты дважды. Повторяющиеся задержки доступа, нехватка сопровождения, сбои платформы или трудности переноса становятся общими работами с владельцем и сроком. После изменения проверяют эффект на следующих волнах. Результат — решения по ограничениям предприятия, основанные на данных проектов.

Подведение итогов. Формируют сводку достигнутого эффекта, охвата, общих затрат и оставшихся проблем. Отдельно показывают команды, перешедшие к Пилоту и завершившие его, достигнутые цели и планы их достижения. Указывают ограничения сравнения и случаи, в которых подход не сработал. Материалы служат основанием для решения о продолжении программы и становятся исходной точкой следующего периода, сохраняя историю изменений.

Подробности: гайд процесса, сведение разнородных результатов, показатели предприятия.

В продуктовой команде

Пять процессов ведутся параллельно на всех этапах внедрения. Ниже для каждого показано, что делается на Подготовке, PoC и Пилоте и с каким результатом команда переходит дальше.

Управление внедрением

РП совместно с владельцем продукта и ИИ-инженером организуют внедрение: согласуют цели, обеспечивают участие людей и ресурсы, разрешают зависимости и принимают решения по этапам.

Подготовка. Уточняют, какой поток работ передаётся агентам, что должно измениться в сроках поставки и составе команды, какие требования к качеству и ограничения по рискам сохраняются. Назначают ответственных, выделяют время специалистов. На Подготовке цель формулируют по сути, без формул и численных показателей. На PoC выбирают вычислимые метрики, текущие и предварительные целевые значения; на Пилоте уточняют расчёты и при необходимости ориентиры. Составляют план Подготовки, PoC и Пилота с результатами, зависимостями и контрольными точками. Выбирают задачи для PoC; конкретный поток Пилота уточняется по его результатам. Итог — согласованные цели, план и обязательства сторон, подтверждённые условия начала PoC.

PoC. Организуют проверку рабочего цикла: подключение агентов, простые задачи, затем типичные задачи будущего Пилота. РП обеспечивают ответы специалистов и рассмотрение результатов, разрешают выявленные зависимости и корректируют план. Агент внедрения помогает увидеть недостающие результаты и подготовить сведения; решения остаются у людей. К концу этапа определены проверенная область применения, ограничения, состав участников и условия Пилота. Решение о переходе принимается по доказательствам готовности.

Пилот. Обеспечивают работу команды на согласованном потоке и участие специалистов. По метрикам выявляют ограничения, при необходимости улучшают агентов и способ работы. Если согласованные показатели и остальные условия чек-листа завершения Пилота выполнены, РП фиксирует успешное завершение без отдельного решения Спонсора. Если условия не выполнены, Спонсор решает продолжить Пилот, изменить условия с продолжением либо остановить его. После изменения целей их достижение проверяют при следующем подведении итогов. Завершение Пилота команды само по себе не разрешает масштабирование предприятия.

Подробности: гайд процесса, командный процесс, поэтапное формирование целей и метрик, паспорт проекта.

Подготовка и вовлечение участников

Обеспечивают конкретных исполнителей, время и освоение новых обязанностей. РП организует участие и обучение, владелец продукта выделяет ресурс, ИИ-инженер демонстрирует работу и помогает специалистам. Результат — люди выполняют необходимые действия и получают доступную помощь. Сами правила движения задачи определяются в процессе встраивания агентов.

Подготовка. В паспорте назначают участников и заместителей, записывают выделенное время и порядок обращения за помощью. Организуют ежедневные встречи, еженедельный доклад Спонсору и ответ в течение двух часов рабочего времени. Проводят вводный семинар с предусмотренным учебным упражнением; сохраняют материалы и подтверждения. Итог — участие обеспечено, специалисты познакомились со своими действиями.

PoC. ИИ-инженер ведёт взаимодействие с агентами и на задачах продукта демонстрирует постановку, уточнения, ревью, замечания и исправления. Производственным специалистам доводят новый процесс и разбирают вопросы по их ролям. На поздних задачах они могут подключаться через

инженера, проверяющего материалы в обе стороны. К Пилоту подтверждены обучение, назначения, время и помощь. Самостоятельная работа специалистов с агентами на PoC не является обязательным условием.

Пилот. Специалисты продукта непосредственно выполняют свои обязанности; по фактам работы определяют, кому ещё нужны помощь, обучение или время. Новые участники получают сопровождение. Освоение подтверждают реальными действиями, не присутствием на встрече и не выполнением работы за специалиста инженером внедрения. К завершению закрепляют дальнейшее сопровождение, время и замещения; полной передачи инженерных обязанностей не требуют.

Подробности: гайд процесса, назначение участников, проверка условий команды.

Обеспечение технической готовности

ИИ-инженер вместе с DevOps, владельцами систем и ИБ создаёт условия, в которых агенты могут изменять продукт и проверять результат без персонального ноутбука и скрытых ручных действий.

Подготовка. Заполняют единую карточку инфраструктуры: конфигурация инференса, ИИ-платформы, инструментов конвейера и тестовых сред; размещение систем и сетевая связность. Описывают и предоставляют учётные записи и права людям, системам и ИИ-платформе. Проверяют операции из их фактических рабочих контуров, выполняют сборку, публикацию, деплой и диагностику существующей ревизии. Результаты фиксируют в той же карточке. После подтверждения готовности выбирают задачи PoC в целевом трекере.

PoC. Проверяют путь на реальных задачах, устраняют ручные операции и делают запуски повторяемыми. Среды должны поддерживать минимум сквозные E2E-проверки выбранных сценариев и воспроизводить существенные свойства production. Проверяют сброс данных, изоляцию параллельных работ и сохранение версии для приёмки. По сценариям подключают программно управляемые моки, эмуляторы оборудования, приёмники побочных эффектов и управление временем. К Пилоту доступны подтверждённые среды, команды воспроизведения и диагностика выбранного потока.

Пилот. Поддерживают доступность, мощности и необходимый параллелизм, устраняют сбои и задержки предоставления сред. При расширении потока добавляют данные и механизмы проверок, контролируют изменения платформы, инференса и интеграций. Различия с production и непроверенные свойства остаются видны. Результат — устойчивый рабочий конвейер, в котором агент может проверить и передать результат. Весь продукт заранее перестраивать не требуется, но условия каждого нового сценария обеспечивают до его выполнения.

Подробности: гайд процесса, инфраструктурные работы, тестовые среды и механизмы проверки, ИИ-платформа.

Встраивание агентов в процессы разработки

Определяют и проверяют общий маршрут задачи через людей, агентов и системы: поручение, выполнение, HITL, исправления, проверки и приёмку. Правила сохраняются в ЖЦ задач и Git. Здесь устанавливают необходимые роли, события и условия продолжения; конкретных исполнителей и освоение ими действий обеспечивает подготовка участников.

Подготовка. Описывают действующий ЖЦ задач, ветки, MR, проверки, возвраты и приёмку. После выбора задач проверяют чтение их описаний и вложений агентами. Подробный целевой маршрут заранее не требуется: сведения становятся основой для проб на PoC.

PoC. На пробах уточняют поручение задач, переходы, точки HITL и условия продолжения. ИИ-инженер воспроизводит действия проверяющих; при подключении коллег проверяет обмен в обе стороны. Временные инструменты допустимы, если контакты и решения репрезентативны Пилоту. В целевом контуре проверяют сквозной маршрут, включая исправления, приёмку и передачу человеку; при смене систем повторяют затронутые операции. Итог — описанный и проверенный процесс. Обучение на этой работе не требует самостоятельного ведения задач специалистами на PoC.

Пилот. Поддерживают соответствие ЖЦ задач и Git реальному потоку: связи задач с MR, ревизиями, проверками и решениями, рабочие переходы и передачу человеку. Исправляют потерянные обращения и обходы, актуализируют правила. Изменение области или режима HITL/HOTL проводят по выбранным в процессе эффективности мерам с проверкой маршрута и сохранением необходимого контроля. Освоение действий людьми оценивают отдельно от работоспособности процесса.

Подробности: гайд процесса, ЖЦ задач, правила Git, выбор и проверка улучшений.

Управление эффективностью работы команды и агентов

РП организуют цикл «измерить результат → найти ограничение → внести изменение → проверить эффект». ИИ-инженер и специалисты улучшают агентов, навыки и контекст, взаимодействие людей, процессы и тестовые среды.

Подготовка. В паспорте формулируют содержательную цель, без формул, выбранных метрик и численных ориентиров. Выбор показателей и заполнение плана мониторинга выполняются на PoC по этапному порядку. Работы и результаты PoC берут из чек-листа готовности к Пилоту, сроки — из паспорта. В работах предусматривают пробы, разбор ошибок, доработку и повторные проверки; конкретные пробы и контекст продукта выясняют по ходу PoC.

PoC. Адаптируют агентов к продукту, подключают знания и навыки, разбирают ошибки и проверяют исправления на задачах. Контекст выясняют опытным путём: обращаются к коду, существующим материалам и специалистам по мере появления вопросов, проверенные сведения сохраняют в инструкциях, навыках и памяти агентов. На этом этапе выбирают метрики результата, процесса и баланса, готовят исходную базу и план мониторинга к Пилоту. Настраивают ежедневный сбор сведений и пробную недельную сводку. Проверяют связь событий с задачами, полноту данных, расчёт объёма и времени, обработку повторного сбора без двойного учёта. Расчётные величины отличаются от фактов, отсутствие данных — от нулевых затрат. К Пилоту должны работать исходный набор навыков, сбор сведений и разбор результатов; известны ограничения системы и измерений.

Пилот. Агент ежедневно собирает сведения, вычисляет метрики и готовит недельную сводку. Команда рассматривает динамику, уточняет существенные ошибки и выбранные аномалии расчётов, определяет причины ограничений. Например, ожидание ревью требует изменить доступность людей, пропуски тестов — навыки и проверки, задержка среды — взаимодействие с инфраструктурой. Для улучшения задают ожидаемый эффект, область, исполнителя и срок; после ограниченной проверки сохраняют, дорабатывают или отменяют изменение. Рост автономности проверяют вместе с затратами людей на исправления. Итог — подтверждённые улучшения и основание для следующих изменений; версии навыков, процессов и сред связаны с результатами наблюдения.

Подробности: гайд процесса и циклы PDCA, мониторинг и решения, сбор показателей, План мониторинга и проверка измерений.

По ролям

На уровне предприятия

Спонсор

Утверждает цели и приоритеты внедрения, обеспечивает ресурс и принимает решения о продолжении и масштабе программы.

Прогнозирование эффектов. Инициирует этап и обеспечивает необходимые полномочия и ресурсы. По ходу работы рассматривает промежуточные выводы команды внедрения и согласует ориентиры для дальнейшей проработки. К завершению этапа команда внедрения представляет текущую и предлагаемую оргструктуру, их стоимость, обоснование ожидаемых изменений объёмов и сроков поставки ПО, а также бюджет, сроки и риски трансформации. На итоговой встрече спонсор принимает решение о запуске программы и утверждает выбранный вариант, цели и ограничения либо направляет предложения на доработку или отказывается от внедрения.

Пилотная волна. Утверждает цели, область, ресурс и ответственность за программу. Обеспечивает участие владельцев продуктов и общих служб, разрешает вопросы, выходящие за полномочия руководителя программы. По результатам пилотов определяет, достаточно ли оснований для расширения.

Волны масштабирования. Согласует приоритеты и ресурсы следующих волн, рассматривает достигнутый эффект и общие ограничения. При необходимости меняет темп и область внедрения, обеспечивает решения по межфункциональным препятствиям. Расширение поддерживает с учётом результата и возможностей сопровождения.

Подведение итогов. Сопоставляет достигнутые изменения с целями и затратами, принимает решение о завершении или следующем периоде программы. Обеспечивает владельцев и ресурс постоянной работы, согласует дальнейшие приоритеты и обязательства по недостигнутым целям.

Подробности: гайд роли, владелец программы, комитет внедрения.

Руководитель программы

Координирует внедрения команд, общие зависимости, ресурсы и подготовку решений предприятия.

Прогнозирование эффектов. Координацию первых проб обеспечивает инициатор или назначенный им участник: договаривается о задачах, участии специалистов, сроке и способе фиксации результатов. Сводит наблюдения исследовательской группы и готовит основания решения о программе; при её запуске передаёт материалы руководителю программы.

Пилотная волна. Формирует общий план, согласует состав первых команд, сроки, сопровождение и зависимости от общих служб. Организует подготовку решений и разбор препятствий, получает результаты команд. Локальные планы ведут РП продуктов; руководитель программы обеспечивает согласованность между ними и общими работами.

Волны масштабирования. Готовит состав и ресурс следующей волны, проверяет переносимость опыта и возможность поддержки. Организует рабочие сверки и рассмотрение вопросов комитетом, назначает ответственных за общие действия и проверяет исполнение. По результатам корректирует последовательность подключений.

Подведение итогов. Готовит сводку выполнения программы: результаты, затраты, ограничения, открытые обязательства и предложения. Организует решение о следующем периоде и передачу постоянным владельцам поддержки, инфраструктуры и методологии. Сохраняет ответственных и

сроки для незавершённых работ.

Подробности: гайд роли, организация программы, паспорт программы.

Методолог офиса внедрения

Задаёт единые правила внедрения в рамках корпоративных политик и ограничений: допустимые модели, порядок получения инфраструктуры, обязательные метрики и состав отчётности. Обеспечивает сопоставимость результатов команд и помогает оптимизировать общие расходы предприятия.

> *Если участвует команда внедрения*
>
> **Специалист внедрения.** Готовит предложения по моделям, конфигурациям и метрикам, методические материалы и обучение на основе опыта проектов.
>
> **Методолог офиса внедрения.** Закрепляет согласованные требования предприятия в общих материалах, проверяет их применение в командах и сопоставимость отчётности. Организует разбор исключений и согласование изменений с уполномоченными участниками предприятия.

Пилотная волна. Описывает единые требования для первых внедрений: допустимые модели и варианты подключения инференса, порядок получения инфраструктуры, обязательные метрики, правила расчёта и представления результатов. Использует действующие корпоративные политики и согласованные технические решения. Проверяет, что планы команд учитывают эти требования; помогает применять методологию и разбирать сложные случаи. Эти функции может выполнять назначенный методолог без отдельного офиса.

Волны масштабирования. Проверяет применение общих требований при подключении команд и сопоставимость их результатов. Сводит потребности в моделях, инференсе и инфраструктуре, выявляет дублирование затрат и готовит предложения по общим решениям. Ведёт версии правил, согласованные исключения и изменения. Организует обучение и перенос проверенных практик; различия продуктов отражаются в адаптациях, сохраняя обязательные требования предприятия.

Подведение итогов. Сводит сопоставимые результаты и общие расходы, фиксирует действующие требования, исключения и проверенные практики. Готовит предложения по изменению корпоративных правил и общих сервисов. Закрепляет порядок обновления материалов, поддержки команд и обучения после завершения программы.

Подробности: гайд роли, офис внедрения ИИ.

В продуктовой команде

ИИ-инженер

Обеспечивает рабочий цикл агентов, адаптирует навыки и знания, настраивает сбор сведений и проверяет улучшения.

> *Если участвует команда внедрения*
>
> **Специалист внедрения.** ИИ-инженер внедрения первоначально настраивает и адаптирует агентов, разбирает сложные случаи, настраивает мониторинг и обучает ИИ-инженера продукта на задачах проекта.
>
> **Специалист продукта / предприятия.** ИИ-инженер продукта осваивает диагностику, улучшение навыков и сопровождение агентов и принимает согласованные работы по мере готовности. Часть работ может оставаться у специалиста внедрения.

Подготовка. Вместе со специалистами обследует продукт, конвейер, контекст, инфраструктуру и доступы. Определяет условия первых задач и недостающие средства, помогает выбрать задачи PoC и спланировать проверки. Готовит рабочее окружение и технические подключения; сведения о действующих процессах собирает для последующей адаптации агентов.

PoC. Адаптирует инструкции и навыки к продукту, подключает контекст и ведёт задачи: сначала минимальный рабочий путь, затем простые и целевые изменения. Разбирает ошибки, уточняет навыки и инструменты, проверяет исправления. Настраивает ежедневный сбор сведений и пробные сводки, фиксирует область применимости и готовит команду к Пилоту.

Пилот. Сопровождает поток, который ведёт продуктовая команда: помогает при сбоях и сложных случаях, развивает навыки, контекст и технические средства. Проверяет эффект изменений, поддерживает сбор показателей и помогает разбирать аномалии. Постепенно передаёт инженерные обязанности специалистам продукта с практической проверкой освоения; необходимое сопровождение сохраняется по договорённости.

Подробности: ИИ-инженер.

РП

РП внедрения организует работу сопровождения и общие материалы внедрения; РП продукта обеспечивает участие и обязательства продуктовой команды. Они используют согласованные план, встречи и сводки.

> *Если участвует команда внедрения*
>
> **Специалист внедрения.** РП внедрения организует работы выделенной команды внедрения, готовит план и отчётность по этим работам.
>
> **Специалист продукта / предприятия.** РП продукта организует участие специалистов продукта: выделение времени, выполнение работ, ответы агентам и проверки результатов. Оба используют общий план и встречи; у каждой работы один ответственный. При совмещении назначений эти обязанности выполняет один человек.

Подготовка. Вместе с владельцем продукта и ИИ-инженером уточняет цели, область, сроки и необходимые ресурсы; оформляет содержательную цель в паспорте без формул и численных показателей. Назначает участников своей стороны, обеспечивает их доступность и выполнение подготовительных работ. Согласует общий план, контрольные точки и порядок решения вопросов; организует подтверждение условий начала PoC.

PoC. Обеспечивает готовность задач, участие специалистов, ответы на вопросы и рассмотрение результатов. Разрешает организационные зависимости, уточняет план по выявленным ограничениям и готовит решение о переходе к Пилоту. РП внедрения представляет продвижение и результаты; РП продукта обеспечивает действия своей команды.

Пилот. Организует недельный разбор показателей, проверку выделенных аномалий и выполнение улучшений. Работает с задержками участия людей и обеспечением ресурсов, готовит итоговое решение и дальнейший план. Не требует отдельного ручного учёта каждого действия: использует сведения из рабочих систем и уточнения существенных расхождений.

Подробности: РП и распределение работ между участниками.

Владелец продукта

Определяет продуктовую ценность, приоритеты и границы полезного результата, принимает решения по поведению и объёму работ.

Подготовка. Выбирает полезную область внедрения и задачи для PoC, формулирует ожидаемый результат и критерии продуктовой проверки. Обеспечивает доступ к предметным специалистам и согласует, какие работы и риски входят в первоначальную область. Конкретный поток Пилота уточняет по результатам PoC.

PoC. Отвечает на продуктовые развилки, уточняет границы требований и рассматривает полученные результаты вместе с заключениями специалистов. Помогает подобрать простые и типичные задачи, по которым можно проверить применимость подхода. К Пилоту согласует область регулярной работы и порядок продуктовой приёмки.

Пилот. Обеспечивает приоритетный поток задач, принимает результаты либо указывает конкретные отклонения от ожиданий. При расширении выбирает следующие виды полезных работ и обеспечивает участие экспертов. Участвует в оценке эффекта и решениях о продолжении; техническую достаточность проверок определяют профильные специалисты.

Подробности: владелец продукта.

Аналитик

Обеспечивает непротиворечивое описание поведения продукта, достаточное для реализации и проверки.

> *Если участвует команда внедрения*
>
> **Специалист внедрения.** Аналитик внедрения помогает упорядочить требования, подготовить постановки для агентов и наладить работу с аналитическими материалами.
>
> **Специалист продукта / предприятия.** Аналитик продукта уточняет предметные правила и ожидаемое поведение, проверяет постановки и ответы агентов на соответствие замыслу. Эти работы можно поручить аналитику внедрения, если ему доступны необходимые знания и полномочия.

Подготовка. Находит действующие требования и спецификации, предоставляет предметный контекст и примеры для первых задач. Вместе с владельцем продукта уточняет ожидаемое поведение, границы и открытые вопросы; определяет правила ведения аналитических материалов и пробелы, существенные для выбранной области.

PoC. Предоставляет предметные сведения ИИ-инженеру; на последних задачах через инженера разбирает неоднозначности и варианты поведения, помогает уточнять и декомпозировать задачи. Проверяет требования и аналитические результаты, передаёт повторяющиеся ошибки ИИ-инженеру для улучшения контекста и навыков. Если аналитическая работа включена в область агента, подтверждает её применимость на реальных случаях.

Пилот. Готовит или проверяет требования в согласованном потоке, отвечает на вопросы и фиксирует принятые предметные решения. При подключении новых типов задач уточняет знания и правила проверки. Участвует в разборе возвратов и задержек, вызванных неполными или противоречивыми постановками.

Подробности: аналитик.

Разработчик

Обеспечивает инженерные правила и содержательное ревью кода и технических проверок в своей области.

> *Если участвует команда внедрения*
>
> **Специалист внедрения.** Разработчик внедрения адаптирует сборку и проверки, добавляет механики управляемого тестирования и помогает разбирать ошибки агентов.
>
> **Специалист продукта / предприятия.** Разработчик продукта проверяет изменения агентов с учётом устройства и правил кодовой базы и выполняет согласованные действия по интеграции. Ревью и интеграцию можно поручить разработчику внедрения при достаточном знании продукта и правах в репозитории.

Подготовка. Предоставляет правила репозитория, значимые примеры кода, команды сборки и тестов. Согласует порядок ревью и интеграции, определяет проверяющих и владельца слияния изменений. Помогает выявить технические условия первых задач и пробелы в автоматических проверках.

PoC. Предоставляет сведения о коде и правилах разработки ИИ-инженеру. На последних задачах подключается к проверке изменений и MR после предварительной проверки инженером; обратная связь возвращается агентам через инженера. Формулирует конкретные замечания, проверяет исправления и достаточность unit- и интеграционных тестов. Повторяющиеся ошибки разбирает с ИИ-инженером и помогает уточнять инструкции агента.

Пилот. Получает результаты по рабочему маршруту команды, проводит предусмотренное ревью и участвует в интеграции. Для новых или рискованных областей определяет необходимую глубину проверки. Предлагает изменения контроля на основании доказательств и наблюдает последствия; смена этапа сама по себе не ослабляет ревью.

Подробности: разработчик.

Архитектор

Определяет архитектурные ограничения и рассматривает изменения с существенным влиянием на продукт и его связи.

> *Если участвует команда внедрения*
>
> **Специалист внедрения.** Архитектор внедрения обследует решение и предлагает изменения интеграций, тестовых сред и технических ограничений для автономной работы агентов.
>
> **Специалист продукта / предприятия.** Архитектор продукта проверяет совместимость предложений с архитектурой и планами развития продукта. Право утверждать решения закрепляют в проекте; его можно делегировать архитектору внедрения.

Подготовка. Объясняет устройство затронутых компонентов, границы ответственности, правила интеграции и значимые ограничения. Помогает определить область первых задач и случаи, требующие участия эксперта. Передаёт ИИ-инженеру достаточный технический контекст для последующей адаптации агентов.

PoC. Рассматривает спорные решения и задачи с архитектурными последствиями, проверяет соответствие существующему устройству продукта. Уточняет обнаруженные пробелы контекста и разделение типовых изменений и исключений. Помогает определить границы применимости навыков и условия работы с задачами PoC.

Пилот. Подключается к изменениям в своей области ответственности, рассматривает новые интеграции и расширение охвата. Вместе с разработчиком определяет, где допустим менее блокирующий контроль, а где сохраняется подробное рассмотрение. Анализирует повторяющиеся отклонения и предлагает изменения архитектурных правил или технических средств.

Подробности: архитектор.

Тестировщик

Определяет достаточность проверок и доказательств, проверяет поведение продукта и качество результата.

> *Если участвует команда внедрения*
>
> **Специалист внедрения.** Тестировщик внедрения помогает внедрить автотесты, подготовить данные и специальные механики: программно управляемые моки, виртуальное время, эмуляторы и сброс состояния. Реализацию распределяют между ним, разработчиком и DevOps.
>
> **Специалист продукта / предприятия.** Тестировщик продукта определяет предметные сценарии, проверяет достаточность покрытия и результаты тестов агентов. Специалист внедрения может выполнять и эти работы по мере освоения продукта и получения полномочий.

Подготовка. Уточняет критерии проверки первых задач, необходимые среды, данные и формат доказательств. Вместе с разработчиком определяет применимые виды тестов и недостающие средства. Выделяет, какие проверки доступны агенту и какое участие человека необходимо для ближайшей работы.

PoC. Помогает ИИ-инженеру уточнять сценарии и критерии проверок. На последних задачах рассматривает автотесты и отчёты, подготовленные к проверке инженером, подтверждает воспроизводимость сценариев; замечания возвращаются агентам через инженера. Проверяет не только успешный запуск теста, но и то, что сценарий обнаруживает нужную ошибку. Выявленные пропуски и повторные возвраты разбирает с ИИ-инженером и помогает улучшать навыки и проверки.

Пилот. Выполняет согласованную проверку результатов и доказательств, фиксирует замечания и поздние дефекты. Для новой области определяет необходимое покрытие и сохраняемое участие человека. При переходе к HOTL или автоматическим проверкам проверяет достаточность оснований и наблюдает качество и возвраты, чтобы сокращение ручного контроля не скрывало рост доработок.

Подробности: тестировщик.

DevOps

Обеспечивает воспроизводимые сборку и развёртывание, рабочие среды и диагностику в согласованной области продукта.

> *Если участвует команда внедрения*
>
> **Специалист внедрения.** DevOps внедрения развёртывает ИИ-платформу, временные Git, трекер и тестовые среды для PoC, подключает инференс и настраивает интеграции.
>
> **Специалист продукта / предприятия.** DevOps продукта или предприятия предоставляет ресурсы, сетевую связность и доступы, обеспечивает работу систем в инфраструктуре предприятия. Если ресурсы предоставляет один человек, а установку выполняет другой, в плане выделяют отдельные работы с одним ответственным за каждую.

Подготовка. Вместе с ИИ-инженером обследует конвейер, обеспечивает доступы, сборку, хранение артефактов и развёртывание указанной версии. Определяет среды разработки и приёмки, данные, изоляцию и способы восстановления. Планирует недостающие моки, эмуляторы и управление временем до зависимых сценариев.

PoC. Подтверждает сборку и развёртывание в первом рабочем пути, затем проверяет условия задач PoC. Обеспечивает повторяемый сброс, изоляцию параллельных работ, диагностический доступ и перенос выбранной версии в среду приёмки. Устраняет ручные операции и уточняет инструкции управления средами.

Пилот. Поддерживает доступность и параллелизм, реагирует на сбои и задержки конвейера, готовит условия новых проверок. Участвует в улучшениях, сокращающих ожидание сред и ручные действия. При изменении инфраструктуры проверяет влияние на текущую работу и сохраняет возможность воспроизвести результат и диагностику замечаний.

Подробности: DevOps.

Шаблоны

Карточки объектов

> **Что известно.** Карточки содержат актуальные сведения о проекте и условиях работы. Изменения сохраняются в истории документа; секреты в карточки не помещаются.

Паспорт программы предприятия. Содержит цель и охват программы, волны, общие условия, состав команды и ресурсы. Шаблон и описание, пример программы Меридиана.

Паспорт проекта внедрения в продукт. Содержит цель, рамки работ, конфигурацию, календарь, состав команды и выделенное время. Шаблон и описание, пример личного кабинета. Состав команды включён в паспорт; отдельная карточка не требуется.

ЖЦ задачи. Описывает статусы, переходы, условия начала и завершения, ответственных и связь с Git-процессом. Шаблон и описание ЖЦ задач, пример ЖЦ.

Правила Git. Описывает ветки, MR, проверки, исправления и интеграцию; задаёт события ревью, CI и сред для измерения ожиданий. Шаблон и пояснения, пример правил Git.

Тестовые среды. Описывает назначение и адреса DEV/UAT, версии, данные, команды развёртывания, сброса и проверки. Как подготовить среды, приёмка в ЖЦ задач.

Инфраструктура и доступы. Одна карточка описывает конфигурацию инференса, ИИ-платформы, инструментов конвейера и тестовых сред; размещение и сетевую связность. В ней же перечисляются необходимые учётные записи и права людям, системам и ИИ-платформе, затем фиксируются выдача и проверки. Как заполнить карточку, шаблон карточки инфраструктуры, пример.

План мониторинга продукта. Связывает цели с метриками результата, процесса и баланса, исходной базой, ориентирами, источниками и ритмом сбора. Заполняется и проверяется на PoC: метрики, текущие значения и предварительные ориентиры. На Пилоте уточняются расчёты и при необходимости целевые значения. Шаблон плана, как выбирать метрики и заполнять, пример раннего PoC.

Чек-листы

> **Что должно/может быть.** Проверяют условия и достаточность доказательств. Готовность оценивается отдельно для PoC и Пилота; условия зависят от выбранной работы и режима.

Готовность команды. Выявляет пробелы в постановке задач, процессах, проверках и участии людей. Помогает выбрать улучшения; незакрытые пункты не блокируют внедрение целиком. Как использовать результат, Описание и чек-лист условий команды, форма примера.

Готовность инфраструктуры и инференса. Подтверждает необходимые мощности, подключения, доступы, сборку и среды для ближайших сценариев. Проверка выполняется из нужного контура, а не только с компьютера инженера. Инфраструктурные работы, конфигурация платформы и проверка подключений, чек-лист готовности к PoC, чек-лист готовности к Пилоту.

Гейт перехода к PoC. Подтверждает обязательные результаты Подготовки, условия онбординга и первых задач; фиксирует решение и оставшиеся зависимости. Переход к PoC, чек-лист готовности к PoC.

Гейт перехода к Пилоту. Подтверждает результаты PoC, качество выбранного потока, готовность команды, рабочих правил и мониторинга. Переход к Пилоту, чек-лист готовности к Пилоту.

Завершение Пилота. Подтверждает достижение согласованных показателей по проверенным данным, актуальность процессов и инфраструктуры и обеспеченное сопровождение. Успешное завершение фиксирует РП; при невыполнении условий дальнейшие действия определяет Спонсор. Чек-лист завершения Пилота и работы этапа.

Todo-листы

> **Что нужно сделать.** Планы задают действия, ожидаемый результат, ответственного, зависимости и срок. Работы, уже заведённые в трекере, подключаются ссылками.

План развёртывания инфраструктуры и подключения инференса. Определяет, что установить или подключить, какие ресурсы и доступы получить и как подтвердить готовность. Конфигурация, размещение и подключения.

Работы PoC. Перечень работ и ожидаемых результатов берётся из чек-листа готовности к Пилоту; конкретные исполнители и действия назначаются в трекере. Сроки и контрольные встречи задаются в паспорте проекта. Отдельный документ плана не требуется.

Работы Пилота. Сроки, область, команда и ресурсы задаются паспортом проекта, метрики и ориентиры — планом мониторинга. Работы берутся из чек-листа завершения Пилота, конкретные назначения и ближайший объём ведутся в трекере. Недельный разбор определяет следующие действия; отдельный документ плана внедрения не требуется.

Журналы и протоколы встреч

> **Что происходило.** Записи собираются из доступных систем и связываются с первичными данными. Команда проверяет существенные выводы и уточняет сведения; повторный ручной ввод тех же фактов не нужен.

Метрики — еженедельно. Журнал содержит значения показателей, вычисленных по указанной версии плана мониторинга, с периодом, источниками и ограничениями. Как вести журнал и шаблон, форма примера.

Журнал решения задач — ежедневно. Показывает фактические действия и результаты за день, участие людей и задержки. Даёт ссылки на задачи, MR, проверки и сообщения. Ежедневный сбор, описание и шаблон журнала задач, форма примера.

Журнал улучшений агентов, процессов, инструментов и тестов — ежедневно. Сохраняет выполненные изменения, ожидаемый эффект, способ проверки и полученный результат. Связывает изменение показателей с конкретной работой и фактическими исполнителями; для работы ИИ-инженера внедрения сохраняет причину, почему её не выполнял инженер продукта. План мониторинга и проверка измерений, описание и шаблон журнала улучшений, форма примера.

Протоколы встреч. Сохраняют принятые решения, их основания и согласованные действия с ответственными и сроками. Связывают договорённости с рассмотренными материалами и результатами внедрения. Как проводить недельный разбор, итоги и план недели.

Практическая организация рабочих материалов и помощи агента: агент и репозиторий внедрения.

НА ПРЕДПРИЯТИИ

Управление целями и приоритетами внедрения

Процесс определяет, какое изменение производства ПО нужно предприятию, какие ожидания подтверждены и что важнее при ограниченных ресурсах. Прогнозирование эффектов — начальный этап, а управление целями продолжается по мере получения результатов продуктов.

1. Участники, входы и результат

Спонсор инициирует этап и рассматривает предложения; команда внедрения готовит анализ с предметными специалистами. После запуска программы руководитель программы организует актуализацию целей и приоритетов, владельцы продуктов подтверждают условия своих областей.

Входы — потребности рынка, бэклоги, текущая структура и стоимость производства, ограничения и доступные возможности. Результат — решение и актуальные цели, область и ограничения в паспорте программы. Прогноз, оценка и полученный эффект различаются.

2. Прогнозирование эффектов

Спонсор инициирует ограниченный этап без обязательного предварительного создания офиса и полной программы. В ходе проб команда внедрения изучает реальные задачи, участие людей и ограничения; существенные развилки рассматриваются по мере появления, а не только в финале.

К итоговой встрече готовится предложение: текущая и предполагаемая оргструктура и стоимость, ожидаемые изменения поставки, необходимые ресурсы, бюджет и сроки трансформации, новые риски и основания прогноза. Указываются диапазоны и допущения, если точных данных нет. Одна удачная задача не доказывает эффект всех продуктов.

На встрече по презентации Спонсор принимает решение о программе, продолжении ограниченных проб либо отказе. **Выход этапа — решение**, а не оставленное на неопределённое обдумывание предложение. Основания и условия решения сохраняются в доступных записях; до программы достаточно существующих рабочих систем.

3. Пилотная волна

Руководитель программы переводит принятое направление в область первых внедрений, приоритеты и доступный ресурс. Владельцы продуктов объясняют, какое изменение нужно их потоку. Предприятие не задаёт всем одинаковый процент ускорения независимо от исходных условий.

В каждом продукте цель развивается по этапному порядку: Подготовка — по сути, PoC — вычислимые метрики, текущие значения и предварительные ориентиры, Пилот — уточнение расчётов и проверка результата. Числовой прогноз программы не подменяет эту работу команды.

Раз в согласованный период рассматриваются факты, новые ограничения и конфликт приоритетов. Предложение изменить цель содержит основание, прежнее и новое состояние, влияние на продукты и ресурсы. Спонсор принимает решения в своей зоне; локальные уточнения методики ведутся по полномочиям проектов с сохранением сопоставимости.

4. Последующие волны и готовность к итогам

Для утверждённых волн проверяется, сохраняются ли исходные предположения и приоритеты. Результат первого продукта используется с описанием условий; он не превращается в обещанный эффект всей компании. Решения о составе продуктов принимает процесс состава и волн, о ресурсах — управление программой.

К подведению итогов должны быть доступны версии целей, прогнозы, принятые изменения и подтверждённые результаты. Достигнутые показатели отделены от планов достижения и неподтверждённых ожиданий. Подробные работы Масштабирования и финального этапа здесь не планируются; отдельный гейт предприятия не вводится.

5. Управленческий цикл

Запуск: новое основание прогноза, результат продукта, изменение внешних условий или конфликт приоритетов. **PLAN:** выбрать проверяемое ожидание и последствия решения. **DO:** получить ограниченные наблюдения или результаты текущих внедрений. **CHECK:** сопоставить их с допущениями и ограничениями. **ACT:** сохранить цель и приоритет либо согласованно изменить их, область или ресурс.

Итог — запись решения и обновлённый паспорт, а не новый независимый список поручений. Подготовка сводных доказательств — в сведении результатов; расчёты — в метриках предприятия.

Методология и передача опыта

Процесс поддерживает единые корпоративные требования и позволяет командам использовать проверенный опыт на собственных продуктах. Его результат — применимые версии правил и навыков, освоенные действия и учтённые исключения. Одинаковый файл инструкций не доказывает одинаковой пригодности для разных продуктов.

1. Участники и основные источники

Методолог офиса внедрения ведёт общие требования, сопоставимость и поддержку их применения. Руководитель программы обеспечивает ресурс общих работ. ИИ-инженеры и профильные специалисты проверяют техническое применение; команды продуктов сохраняют свои решения и ответственность.

Исходная методология и её версия, корпоративные ограничения, результаты продуктов и их улучшений — входы процесса. Общие правила сохраняются с версиями и областью применения; проектные документы и адаптации — в репозитории внедрения. Повторные действующие копии не создаются.

2. Прогнозирование эффектов

Отдельный офис и методолог не являются условиями первых проб. Команда внедрения использует доступную методологию, сохраняет факты, ограничения и источники решений. Подтверждённый частный способ ещё не становится обязательным стандартом предприятия.

К решению о программе указываются потребности в методической поддержке и обучении, которые обнаружились в пробах. Это позволяет выделить конкретную функцию, а не заранее создать полный офис без понимания нагрузки.

3. Пилотная волна

Методолог с уполномоченными участниками закрепляет допустимые модели и варианты инференса, порядок получения инфраструктуры, обязательные метрики и состав отчётности. Используются действующие политики предприятия. Предложения команды внедрения не становятся корпоративным правилом без соответствующего решения.

Вместе с РП продуктов проверяется, как эти требования отражены в паспортах, инфраструктурных карточках, процессах и планах мониторинга. Адаптация объясняет различие продукта, область действия и последствия, а не молча снимает обязательную проверку.

Обучение строится на задачах продукта. На PoC инженер показывает работу и сопровождает предметных специалистов, на Пилоте они действуют непосредственно. Процесс освоения практик задаёт эту последовательность; методолог помогает использовать материалы и разбирать общие затруднения, не требует самостоятельной работы уже на PoC.

4. Переиспользование опыта в последующих волнах

Из журналов улучшений выделяется применимый результат: какая проблема была, что изменили, на каких версиях и задачах проверили, какое участие людей осталось и где вывод ограничен. Перед использованием в другом продукте сравниваются стек, зависимости, полномочия, сценарии и режим контроля.

Нужные проверки адаптации выполняет назначенный участник. Нельзя объявить улучшение общим по числу исправленных инструкций или распространить снижение HITL на новую область без доказательств. Пример полезен вместе с условиями, а не только текстом удачного промпта.

Руководитель программы выделяет ресурс на выбранные общие улучшения. Выполненные изменения отражаются с фактическими исполнителями; для инженера внедрения указывается, почему работу не выполнял инженер продукта. Полная передача сопровождения не является обязательной целью процесса.

5. Ритм, исключения и готовность к итогам

В ритме программы разбираются повторяющиеся вопросы и результаты применения правил. Срочное исключение направляется уполномоченному участнику по мере появления; до решения действует разрешённая область. Для исключения сохраняются причина, ограничения, срок пересмотра и решение.

К подведению итогов доступны действующие версии требований, исключения, проверенные примеры и назначенная поддержка их обновления. Подробные работы финального этапа и Масштабирования не создаются в этом гайде.

Связи: офис внедрения, общая инфраструктура, сведение результатов, журнал улучшений.

Управление составом и волнами внедрения

Процесс выбирает, какие продукты подключаются и в какой последовательности. Программа с самого начала задаётся на уровне предприятия; она не возникает автоматически из масштабирования результата отдельного продукта. Каждая подключаемая команда проходит собственные Подготовку, PoC и Пилот.

1. Ответственность и источники

Руководитель программы готовит состав и порядок подключений с владельцами продуктов. Спонсор принимает решения по области и существенным приоритетам. Методолог проверяет применимость общих требований, профильные участники — доступный ресурс.

Источник состава и календаря — паспорт программы. Подробные работы остаются в проектах продуктов; их не копируют в ещё один общий трекер задач.

2. Прогнозирование эффектов: выбрать репрезентативные пробы

Составить краткое сопоставление кандидатов: объём разработки функциональности, достаточность бэклога, коммуникационные зависимости, срочность, доступность команды и технических условий. Приоритет — продукт с потоком фич на несколько месяцев и возможностью детализировать, реализовать и проверить их.

Проект, где основная работа — переговоры со многими подразделениями, совместные планы и несколько настроек после долгого согласования, не даёт того же основания для агентского Пилота. Он может быть важен бизнесу, но требует другого устройства работы людей. Классификация и признаки — в выборе продуктов.

Пробы должны дать основание решения, а не показать заранее гарантированную демонстрацию. При этом срок задачи позволяет пройти несколько раундов замечаний. После первого MR не следует забирать её только потому, что для срочной поставки не оставили времени на исправления.

3. Пилотная волна

После решения о программе согласуются первые продукты, их последовательность, команды и доступные общие мощности. Старт продукта разрешается по его готовности, а не по месту в презентации. Общие зависимости — получение инфраструктуры, обучение, занятость инженеров — видны руководителю программы.

У каждого продукта есть свой паспорт, область и контрольные точки. Ограничения и результаты проб передаются со ссылками, но гейты не наследуются. Если продукт существенно отличается по стеку, зависимостям или организации, это отдельная проверяемая адаптация, а не обязательство повторить все прежние действия буквально.

На сверке рассматриваются задержки, подтверждённые результаты и необходимость изменить порядок подключений. Основание переноса и его влияние сохраняются; продукт не исчезает из области программы ради улучшения показателя охвата.

4. Условия следующих волн

Для предложения следующей волны нужны: область и приоритеты, доступные исполнители, условия инфраструктуры, подтверждённые ограничения и подходящий опыт предыдущих продуктов. Успешный Пилот одного продукта не является единственным достаточным условием подключения остальных.

Гайд описывает критерии выбора и сопровождения утверждённых волн; их подробные задачи не создаются здесь. Решение о существенном изменении состава связывается с приоритетами и ресурсом программы. По данным уточняется ожидаемый объём поддержки, не объявляется универсальная переносимость навыка.

5. Готовность к подведению итогов и ритм

К итогам сохраняется история состава: какие продукты планировались, что подключено, кто завершил PoC и Пилот, какие подключения изменены и почему. Это позволяет отличить результат от изменения знаменателя и неудачный Пилот от ещё не начатого.

Сверка ведётся в ритме программы и перед решением о составе. Исключения рассматриваются при появлении, не ждут формальной очередной волны. Метрики и сведение — по процессу результатов, ресурсные зависимости — по управлению программой.

Управление программой и ресурсами

Процесс обеспечивает выполнение согласованной программы: полномочия, участие производственных команд, ресурс общих работ, управление зависимостями и принятыми решениями. Его результат — исполняемые обязательства и доступные условия, а не только общий календарь.

1. Участники и границы

После решения о программе руководитель программы организует координацию, Спонсор решает вопросы его полномочий. РП продуктов обеспечивают локальные работы; методолог поддерживает единые требования. Подготовку и выполнение части работ может вести внутренняя или профессиональная команда внедрения.

Владельцы продуктов и производственные специалисты сохраняют решения и действия своей зоны. Если цепочка Спонсор → руководитель программы/РП → ключевые специалисты не действует, внешняя помощь не запускает внедрение вместо предприятия.

2. Прогнозирование эффектов

Для ограниченных проб назначаются участвующие специалисты, доступное время и способ решения вопросов. Полная структура офиса, паспорт программы и регулярная корпоративная отчётность не являются условием начала. Команда внедрения готовит наблюдения и предложение трансформации, Спонсор в ходе этапа рассматривает развилки и принимает итоговое решение на встрече.

Полученные подходящие материалы переиспользуются при запуске программы с проверкой применимости. Они не копируются в новый комплект только ради формального начала этапа.

3. Пилотная волна: сделать обязательства исполнимыми

Руководитель программы с владельцами продуктов и РП фиксирует в паспорте программы область, календарь, роли и доступный ресурс. У продуктов — свои паспорта. Общий специалист не получает один и тот же резерв времени повторно в нескольких проектах.

Распределяются работы общей инфраструктуры, методологии и сопровождения. Для каждой конкретной работы в трекере один ответственный, вклад соисполнителей и проверяемый результат. Права выделять ресурсы не возникают у консультанта по факту знания платформы.

Общая сверка рассматривает: что готово для ближайших продуктов, где межпроектная зависимость, какое решение не могут принять команды, что просрочено и кто выполняет следующий шаг. Ответственный получает конкретный вопрос, варианты и срок контроля. Не создаются дополнительные обязательные встречи, если эту функцию выполняет существующий формат предприятия.

4. Сопровождение утверждённых волн

Программа учитывает реальную нагрузку поддержки, очереди получения инфраструктуры и доступность специалистов. Если ресурс ограничивает подключение, выбирается изменение очередности, выделение мощности, сужение области или иной обоснованный вариант. Намерение «ускорить команды» без доступного ресурса не считается решением.

Существенное изменение целей, области, бюджета или календаря проходит согласование по полномочиям и сохраняет историю. Локальные изменения процесса не переносятся на все продукты автоматически. Обязательные корпоративные правила применяются вместе с согласованными исключениями.

Подробные работы Масштабирования здесь не планируются. Гайд задаёт порядок рассмотрения зависимостей уже принятой программы; отдельного нового гейта предприятия нет.

5. Ритм и доказательства управления

РП ведут ежедневную координацию и недельный разбор в продуктах; руководитель программы сводит общие вопросы в утверждённом календаре. Комитет рассматривает повестку решений, а не принимает заново каждое действие команды. Порядок офиса и начальные ориентиры встреч — в материале об офисе.

Решение сохраняет основание, область действия, исполнителя, срок и следующий контроль. Статус «в работе» не подтверждает устранение блокера; нужны полученный ресурс, выполненная операция или изменённое обязательство. Агент помогает подготовить сведения, полномочия остаются у участников.

6. Готовность к подведению итогов

Должны быть доступны текущая версия программы, подтверждённые результаты продуктов, фактические общие расходы, история решений и оставшиеся обязательства с исполнителями. Поддержка общих решений и проектов имеет ресурс после окончания календарного периода. Это состав входных материалов финального этапа; его подробные работы здесь не создаются.

Связи: цели и приоритеты, состав и волны, общая инфраструктура, сведение результатов.

Сведение результатов и общие улучшения

Процесс превращает разные результаты продуктов в основания решений предприятия. Он сохраняет различия условий и отделяет продвижение внедрения, производственный эффект, расходы и прогноз. Его результат — сопоставимая сводка, ограничения выводов и выполненные решения по общим препятствиям.

1. Участники, входы и границы

Методолог обеспечивает определения и сопоставимость, руководитель программы организует получение сведений и общие действия. РП продуктов подтверждают локальные результаты, ИИ-инженеры — происхождение и расчёты. Агент может собирать и сводить сведения по настроенному навыку, но не подменяет отсутствующие факты.

Источники — версии паспортов и планов мониторинга, журналы показателей, задач и улучшений, условия инфраструктуры и записи решений. Формулы и определения остаются в метриках, организация офиса — в howto.

2. Прогнозирование эффектов

Для первых проб достаточно записи исходных условий, наблюдений, участия людей, ограничений и оснований прогноза. Регулярная отчётность предприятия и офис не требуются. Одна задача не доказывает устойчивую экономию, а гипотеза о будущей структуре не называется достигнутым результатом.

Эти сведения входят в предложение трансформации и решение Спонсора о программе. Сохраняются источник и способ оценки; неизвестное значение не заменяется нулём.

3. Пилотная волна: получить сопоставимую первую сводку

1. По каждому продукту получить область, этап, период/когорту, версию методики и конфигурации, значения и ограничения данных. Проверить источник оценок и их подтверждение РП.
2. Сопоставить границы: начало/конец срока поставки, состав человеческого труда, оценка объёма, режим контроля и обязательные проверки. Несопоставимые группы показывать отдельно.
3. Показать прогресс внедрения: число и долю команд, завершивших PoC и перешедших к Пилоту, и число и долю успешно завершивших Пилот. Основания — фактические записи по чек-листам, состав программы и дата среза. Изменение области программы видно вместе с версией знаменателя.
4. Показать производственные результаты по сопоставимым группам, качество, нагрузку людей и расходы. Медианы пересчитать по наблюдениям; доли — по числителю и знаменателю. Проценты улучшений разных продуктов не усреднять как эффект предприятия.
5. Отделить локальные затраты от расходов офиса, общих мощностей и сопровождения. Одни общие расходы не учитываются повторно у каждого продукта.

Неудачные, остановленные и незавершённые внедрения сохраняются в общей картине. Переход к Пилоту не равен его успеху. Итог команды не переопределяется только ради общего красивого отчёта.

4. Общий цикл улучшений

Запуск: повторяющееся препятствие нескольких продуктов или существенное расхождение общей сводки. **PLAN:** выбрать группу причин и ожидаемое изменение, назвать затронутые продукты и проверку. **DO:** выполнить ограниченное изменение общего инструмента, инфраструктурного условия, правила или поддержки. **CHECK:** сопоставить наблюдения и последствия по качеству, человеческому труду и расходам. **ACT:** закрепить, доработать или отменить решение.

Общий недостаток доступа или очередь ресурса не перекладывается на локальное улучшение навыка агента. Частное замечание к одному продукту не превращается автоматически в корпоративный стандарт. Конкретная работа имеет одного ответственного, фактических исполнителей и срок проверки; результаты остаются в рабочих записях программы и связанных проектах без нового обязательного журнала.

5. Ритм и готовность к итогам

Продукты дают сведения в согласованном ритме, опираясь на ежедневный сбор и недельные сводки. На уровне программы рассматриваются общие причины и варианты решений. Подробный поток задач не пересказывается повторно на комитете.

К подведению итогов нужны текущий состав программы, подтверждённые переходы команд, сопоставимые результаты и расходы, история изменений методики и целей, ограничения и оставшиеся обязательства. Достигнутые значения и планы достижения показаны раздельно. Подробные работы финального этапа и новый гейт предприятия здесь не вводятся.

При недостоверных данных назначается уточнение источника или оценки; до него вывод ограничивается. Порядок выбора дальнейших приоритетов — в управлении целями, исполнение общих решений — в управлении программой.

Общая инфраструктура и условия доступа

Процесс обеспечивает продуктам допустимые модели, инференс, ИИ-платформу, ресурсы сред и способы подключения к конвейеру. Он устраняет общие препятствия автономной работе: человек может пользоваться системой вручную, а агенту нужны разрешённые программные операции, связность и диагностика.

1. Ответственность и результат

Руководитель программы организует получение общих условий и решение зависимостей. Методолог закрепляет согласованные требования предприятия, DevOps и ИИ-инженеры выполняют назначенные технические работы в пределах полномочий. Дополнительные проектные роли администраторов или представителя ИБ не вводятся.

Результат — доступный порядок получения инфраструктуры и проверенные подключения для выбранных продуктов. Состав общих решений фиксируется в программе, конкретные размещение, аккаунты, связи и проверки — в карточках инфраструктуры продуктов.

2. Прогнозирование эффектов

До крупных вложений выясняются ограничения: где допустимы LLM-запросы, какие данные можно передавать, доступны ли программные интерфейсы систем, разрешены ли служебные аккаунты, где можно выполнять агентов и тесты. Для ограниченной пробы достаточно условий её сценария.

Временная инфраструктура допустима, но результаты описываются вместе с отличиями. Возможность работы на временном Git не доказывает доступность целевого Git и корпоративных сетей. К предложению программы прикладываются потребности, доступные варианты и существенные нерешённые ограничения.

3. Пилотная волна

Согласовать общий выбор моделей, способ получения инференса и платформы, границы передачи данных, порядок выдачи ресурсов и сопровождение. Платформа и инференс могут размещаться независимо. Техническая конфигурация выбирается по выбранному решению и корпоративным ограничениям, а не по одному универсальному виду установки.

Определить, какие операции выполняются системами, платформой и людьми. Получить нужные учётные записи и сетевые маршруты, выполнить проверку из реальных контуров. Личный доступ специалиста не подменяет разрешение автоматической интеграции.

Команда продукта описывает свои тестовые среды и необходимые e2e. Если нужны новые механики времени, моков, эмуляторов или изоляции, назначаются конкретные работы до зависимых сценариев. Общая инфраструктура не отменяет условий отдельного продукта.

Изменения корпоративных политик, если они нужны, рассматриваются вне проекта через действующий порядок предприятия. Внедрение работает в принятых границах либо явно ограничивает область; новые полномочия не появляются из записи в инструкции агента.

4. Последующие волны и общие ограничения

Для подключаемых продуктов используется согласованный порядок, но потребность и доступы проверяются в их контексте. По фактической нагрузке оцениваются очереди, доступность мощностей, стоимость и нагрузка сопровождения. Общий сервис может переиспользоваться, если обеспечивает изоляцию и нужные операции; отдельные условия продукта сохраняются.

Повторяющиеся задержки подключения или нехватка ресурса становятся общим вопросом программы. Вариант изменения проверяется на затронутом потоке и фиксируется с периодом действия. Простое увеличение мощности без измеренной причины не объявляется улучшением.

Подробные работы волн здесь не планируются. К итогам должны быть доступны актуальные конфигурации, изменения требований, подтверждения использования, затраты и назначения поддержки. Показатели продуктов сопоставляются с фактической, а не первоначально предполагаемой инфраструктурой.

5. Контроль и взаимодействие

После выдачи доступа или изменения системы проверяется конкретная операция; ежедневно команды разбирают текущие отказы, в общем ритме программы — повторяющиеся системные препятствия. Ожидание получения ресурса и его стоимость отделяются от активной работы и от ожидания человека на HITL.

Технические действия раскрыты в гайде DevOps, настройка агентов — в гайде ИИ-инженера, реализация продукта — в процессе технической готовности. Сведение общих расходов — в процессе результатов.

Встраивание агентов в процессы разработки

Процесс определяет и проверяет, как задача проходит через людей, автономных агентов и рабочие системы до принятого результата. Используется общий ЖЦ задач и Git-процесс: агенту поручают часть задач, он самостоятельно ведёт их, обращается к людям в предусмотренных точках и исправляет замечания. Долю таких задач расширяют постепенно.

1. Границы, источники правил и результат

ЖЦ задач задаёт поручение, состояния, переходы, возвраты, приёмку и передачу человеку; правила Git — ветки, MR, проверки и интеграцию. Это основные места проектных правил. Процесс связывает их в один проверенный маршрут.

Результат: описаны события и условия переходов, необходимые роли и полномочия, точки HITL, предъявляемые материалы и подтверждения выполнения. По сквозной проверке задача проходит согласованный путь; действующие правила соответствуют реальной работе.

Здесь определяется, какая роль нужна в точке контроля и какое действие продолжает работу. Конкретных людей, время, замещения и освоение ими действий обеспечивает подготовка и вовлечение участников. Например, проверка доставки запроса и возврата вердикта подтверждает маршрут; обучение разработчика на этом примере подтверждает другой результат. Одна проба может дать оба подтверждения, отдельное дублирование не требуется.

2. Подготовка: описать действующий процесс

РП с участниками описывает существующие статусы и переходы, разработчик — ветки, MR и проверки, аналитик — требования и уточнения, тестировщик — проверки, приёмку и возвраты. Сведения записывают в ЖЦ задач и правила Git. Подробный целевой маршрут Пилота на этом этапе не требуется.

После выбора задач используется результат технической проверки PREP-17: чтение агентом описаний и вложений из трекера. Детализация постановок и проверка понимания происходят на PoC; готовый доступ не подтверждает достаточности требований. Отдельные карточки методологии на каждую задачу не создаются.

Выход: действующий процесс описан, задачи доступны для чтения. Это основа для проб, а не доказательство готовности всего маршрута Пилота.

3. PoC: проверить маршрут и подготовить его к Пилоту

ИИ-инженер воспроизводит взаимодействия будущего рабочего потока: уточняет понимание задачи, рассматривает MR и тесты, возвращает замечания. Он может выполнять действия проверяющих вместо специалистов продукта. При подключении коллег инженер проверяет материалы в обе стороны. В записях различают проверку сценария инженером и реальное действие профильного участника; два последовательных контакта учитывают раздельно. Обучение специалистов организуется в процессе подготовки участников.

По пробам уточняются способ поручения, состояния, связи задачи с MR и ревизиями, материалы предъявления, необходимые роли в HITL и условия продолжения. Задачу разрешают статусом с условиями выборки либо сообщением уполномоченного участника. Известны задача и версия согласованного объёма; повторное сообщение не создаёт вторую задачу.

Точка участия человека следует либо из task/Git-процесса, либо из навыка агента при вопросе, ошибке или недостатке полномочий. В обоих случаях система отправляет запрос в чат с событием и материалами. Каналы и сохранённые решения доступны агентам; личная договорённость, меняющая выполнение, сохраняется в общей доступной записи.

В целевом контуре проверяют путь: поручение → выполнение → MR → замечание → исправление → тестирование → приёмка. Временные системы PoC допустимы, но контакты людей и агентов и смысл решений должны быть репрезентативны Пилоту. При смене систем повторяют затронутые операции. Отдельно проверяют остановку агента и возможность продолжить сохранённую работу человеком; команда может забрать задачу без обязательного объяснения причин.

Выход: ЖЦ задач и Git актуализированы, маршрут подтверждён сквозным прогоном. Проба HOTL перед Пилотом не обязательна. Такая проверка не вводит требование самостоятельной работы специалистов на PoC.

4. Пилот: поддерживать общий процесс в рабочем потоке

Люди и агенты используют согласованные правила. Для каждой задачи сохраняют связи с MR, проверками, ревизиями и приёмкой. Изменение правил отражают в карточках; ручные обходы и нестандартные пути разбирают, чтобы действующее описание соответствовало реальной работе. Это результат PIL-10, отдельный от освоения обязанностей участниками в PIL-05.

Для каждого контакта сохраняют запрос, нужную роль и фактического участника, ревизию, начало содержательной работы и результат. Очередь человека отличается от его активного труда; draft MR не начинает формальное ревью, первый комментарий не доказывает начало всей проверки, merge не означает выпуск. Открытые обращения и их возраст остаются видимыми.

Передача человеку сохраняет изменения, исход задачи и человеческие затраты. Отсутствие объяснения не исключает передачу из учёта. Повторное поручение того же объёма не создаёт новую успешную попытку ради отчёта.

5. Изменение области и режима контроля

Типы задач и режим проверки меняют раздельно. Для новой области сначала обеспечивают инструменты, данные, критерии и проверку маршрута; успех привычной задачи не разрешает новую автоматически.

Выбор изменения и оценка его эффекта выполняются в процессе эффективности. Специализированная проба PIL-13 относится к процессу эффективности; здесь используют её решение и внедряют выбранное изменение маршрута: для HITL → HOTL или снятия плановой проверки сохраняют необходимые автоматические проверки, асинхронные замечания, условия возврата и полномочия. Профильные специалисты подтверждают достаточность контроля. Уменьшение числа блокировок само по себе не доказывает экономии труда.

6. Поддержание правил и обработка отклонений

PIL-11 подтверждает корректность передачи человеку, PIL-12 — работу цепочки HITL; их результатом не является повторный анализ долей и ожиданий. При выполнении задачи исправляют потерянные обращения, неверные события и адресацию по ролям, неработающие переходы и передачу результата. При отсутствии времени, замещения или навыка подключается подготовка участников; при отказе интеграции — техническая готовность. Назначения не переписывают в карточки вместо паспорта: маршрут ссылается на действующих исполнителей ролей.

Повторяющиеся задержки, передачи и последствия изменения контроля анализируются на общем недельном разборе в процессе эффективности. Здесь исполняют выбранные изменения правил и проверяют их работоспособность. Источники — журнал задач, журнал улучшений и журнал показателей; отдельный процесс «только для агентов» и второй журнал не вводятся.

Работы процесса и подробные действия ролей

Основная принадлежность работы определяется её результатом и записана в чек-листе. Ниже показаны работы этого процесса; ссылки на другие процессы описывают взаимодействие, без повторного назначения той же работы. ID и предшественники находятся в исходной работе.

Подготовка.

- PREP-15. Описать общий процесс задач — ответственный РП.
- PREP-16. Описать общий Git-процесс — ответственный Разработчик.

PoC.

- POC-13. Уточнить общий жизненный цикл задач и Git-процесс — ответственный РП.
- POC-14. Описать точки участия человека — ответственный РП.
- POC-15. Определить проверки и приёмку результатов — ответственный Тестировщик.
- POC-16. Обеспечить передачу задачи от агента человеку — ответственный ИИ-инженер.
- POC-17. Проверить рабочий маршрут в контуре Пилота — ответственный ИИ-инженер.

Пилот.

- PIL-10. Вести общий процесс задач и Git — ответственный РП.
- PIL-11. Обеспечивать передачу задач от агентов людям — ответственный РП.
- PIL-12. Поддерживать работоспособность обращений HITL — ответственный РП.

Управление внедрением в продуктовую команду

Процесс связывает содержательную цель, область работ, доступность людей и ресурсов с проверяемыми переходами между этапами. Его результат — выполненные работы, подтверждённые условия и принятые решения. Плановая дата и количество закрытых задач не заменяют готовность.

1. Назначение и границы

РП организует процесс, владелец продукта определяет продуктовый поток и участие специалистов, Спонсор принимает решения о переходах и дальнейших действиях при невыполнении условий. Технические, предметные и измерительные результаты готовятся в профильных процессах.

Входы — потребность продукта, его текущий процесс, доступные специалисты и ограничения предприятия. Основной документ — паспорт проекта. Работы выводятся из чек-листов и назначаются в трекере; отдельные планы PoC и Пилота не создаются.

2. Подготовка

1. Владелец продукта и РП формулируют нужное изменение производства ПО, границы и ограничения. Спонсор рассматривает предложение. На этом этапе не требуются выбранные метрики и численные ориентиры.
2. РП закрепляет участников, время, сроки этапов и встречи. Сопоставляет работы с реальной доступностью людей. Команда внедрения может готовить и выполнять часть работ, но участие производственников сохраняется.
3. После готовности инфраструктуры владелец продукта с аналитиком выбирает задачи PoC в целевом трекере. Нужен запас на исправления; срочная задача, которую после первого замечания придётся немедленно забрать, не подходит для проверки полного цикла.
4. РП собирает подтверждения гейта к PoC, показывает пробелы и проводит встречу. Спонсор принимает решение на встрече; его условия сохраняются в рабочих записях.

Выход: согласованный паспорт, назначенные работы и подтверждённые условия PoC. Заявка на доступ или пустая карточка не закрывают условие.

3. PoC

ИИ-инженер ведёт пробы и регулярное общение с агентами, а РП обеспечивает предметные решения и устранение зависимостей. После онбординга проверяются простые, затем целевые задачи. Работы продукта и работы по развитию агентов различаются в учёте.

По результатам проб уточняются область Пилота, люди, ресурсы, рабочий контур и ограничения. Профильные процессы готовят целевые доступы, общий маршрут людей и агентов, обучение и проверенный мониторинг. Временные системы PoC допустимы; изменившиеся операции Пилота проверяются отдельно.

РП организует гейт к Пилоту: представляет результаты, готовность и открытые вопросы. Спонсор принимает решение; положительная оценка демонстрации сама по себе не разрешает старт. **Выход:** подтверждённые условия Пилота, актуальный паспорт, план мониторинга и ближайшие назначения в трекере.

4. Пилот

Работу непосредственно ведёт продуктовая команда. РП поддерживает поток задач и доступность специалистов; ИИ-инженер помогает и развивает способ работы. Процесс эффективности обеспечивает наблюдение, улучшения и достоверность данных; здесь организуется выполнение принятых действий и решений.

Ежедневно разбираются препятствия, еженедельно — продвижение, показатели, прошлые изменения и ближайшие работы. РП еженедельно докладывает Спонсору, что изменилось, что просрочено и какие препятствия требуют его решения. Изменение области, ресурса или цели фиксируется в соответствующих документах с основанием; агент не меняет их сам.

К итоговой проверке РП собирает показатели и подтверждения условий завершения. Если они выполнены, фиксирует успешное завершение без повторного принятия Спонсором. Иначе представляет варианты: продолжить, изменить условия с продолжением или остановить. После изменения условий их достижение проверяется вновь; само изменение не закрывает Пилот.

5. Отклонения и решения

- Не хватает человека или времени: изменить назначение, выделить ресурс либо скорректировать ближайшую область; не оставлять обязательство без исполнителя.
- Инфраструктура или предметное решение задерживаются: передать конкретный вопрос уполномоченному участнику и назначить следующий контроль. Перенос обязательной работы не закрывает гейт.
- Условия в исходном прогнозе оказались неверны: сохранить факты, варианты и согласованное изменение паспорта. Нельзя молча исключить неудачные задачи из отчёта.
- Новые проблемы появляются, но показатели достигаются: не требовать пустого бэклога улучшений и нулевого HITL.

6. Результаты и связи

Паспорт задаёт цель, область, календарь и команду; трекер — конкретные работы и зависимости. Журналы задач, улучшений и показателей дают подтверждения по своим назначениям. РП использует эти записи, а не вводит второй ручной учёт.

Работы, относящиеся к другим процессам, сохраняют своих ответственных. Гайд координирует переходы и решения, подробное выполнение описано в гайдах ролей.

Работы процесса и подробные действия ролей

Основная принадлежность работы определяется её результатом и записана в чек-листе. Ниже показаны работы этого процесса; ссылки на другие процессы описывают взаимодействие, без повторного назначения той же работы. ID и предшественники находятся в исходной работе.

Подготовка.

- PREP-02. Оформить цели и границы внедрения — ответственный РП.
- PREP-03. Выбрать и назначить задачи PoC в целевом трекере — ответственный Владелец продукта.
- PREP-04. Согласовать сроки и работы PoC — ответственный РП.
- PREP-28. Подготовить материалы и провести встречу по гейту — ответственный РП.
- PREP-29. Принять решение о переходе к PoC — ответственный Спонсор.

PoC.

- POC-01. Согласовать область и режимы работы Пилота — ответственный Владелец продукта.
- POC-03. Согласовать условия и работы Пилота — ответственный РП.
- POC-04. Подготовить материалы и провести встречу по гейту — ответственный РП.
- POC-05. Принять решение о переходе к Пилоту — ответственный Спонсор.

Пилот.

- PIL-01. Уточнить критерии и период итоговой оценки — ответственный РП.
- PIL-02. Проводить ежедневный и недельный разбор внедрения — ответственный РП.
- PIL-03. Подвести итоги и зафиксировать статус Пилота — ответственный РП.
- PIL-04. Определить дальнейшие действия при невыполнении условий Пилота — ответственный Спонсор.

Управление эффективностью работы команды и агентов

Процесс помогает достигать целей внедрения: обнаруживать потери в работе людей и агентов, выбирать изменения и проверять их эффект. Быстрое выполнение агентом своей части ещё не означает улучшения — затраты могли переместиться в ревью, ручные исправления, ожидания или инфраструктуру.

Основой служат связанные циклы **PLAN → DO → CHECK → ACT**: разбор задачи, недельная работа с группами проблем и отдельный анализ выполнения и ожиданий агентов и людей. Это одна очередь улучшений и общий ритм работы команды, а не шесть отдельных совещаний или журналов.

1. Назначение и границы процесса

Результат процесса — решения о необходимых изменениях, выполненные проверки их эффекта и закреплённый способ работы. Для выбранного изменения должны быть известны причина или проверяемая гипотеза, ожидаемый эффект, один ответственный, срок проверки и итог: закрепить, доработать или отменить.

Эффективность оценивают по согласованным показателям результата, процесса и баланса. Выигрыш в поставке и человеческих трудозатратах рассматривают вместе с качеством и расходами. Сокращение числа обращений к людям само по себе не является целью; сохраняемый HITL допустим, если его трудозатраты учтены и показатели выполняются.

Гайд описывает порядок совместной работы и принятия решений. Как менять навыки и контекст, настраивать среду или проводить предметную проверку, раскрывают гайды соответствующих ролей. Определения показателей и расчёты не повторяются здесь.

2. Как формируются цели и метрики на этапах внедрения

Подготовка: цель по сути → PoC: метрики и предварительные ориентиры → Пилот: уточнение расчётов и проверка результата. Цель не обязана сразу иметь численное выражение: способ её проверки формируется по мере понимания продукта, возможностей агентов и доступных данных.

Подготовка — сформулировать, что должно измениться

В PREP-02 команда формулирует содержательную цель: какое изменение в производстве ПО нужно продукту и зачем. Например, быстрее выпускать функциональность из накопленного бэклога без сопоставимого роста затрат, сохраняя качество. Формулы, выбранные метрики и численные целевые значения на Подготовке не задаются.

РП оформляет предложение с владельцем продукта; спонсор согласует цель и границы. **Результат — содержательная цель в паспорте проекта.** Сроки и контрольные встречи задаются в паспорте, работы PoC — в чек-листе готовности к Пилоту и рабочем трекере. Полностью настроенный мониторинг не является условием начала PoC.

PoC — выбрать, как измерять цель и чего ожидать от Пилота

На PoC выясняют, какие метрики из каталога можно вычислять для выбранного потока: какие события доступны, как связаны задачи, MR, проверки и вмешательства людей, где потребуются оценки. По доступной истории сопоставимых работ определяют текущие значения. Если значение пока оценочное или данных недостаточно, сохраняют происхождение оценки, подтверждение РП и ограничения; ноль вместо неизвестного не подставляют.

РП, владелец продукта и ИИ-инженер выбирают метрики результата, процесса и баланса на Пилот. С учётом текущего состояния и проб записывают **предварительные целевые значения** и допустимые пределы. Это обоснованные ориентиры для проверки, которые на Пилоте могут уточняться. Данные работы ИИ-инженера на PoC не принимаются автоматически за показатели производственного потока.

Результат — заполненный план мониторинга: выбранные метрики, текущие значения, предварительные цели, источники, правила расчёта и ритм наблюдения. К Пилоту проверяют ежедневный сбор, пересборку и пробную недельную сводку. На PoC взаимодействие с агентами ведёт ИИ-инженер; специалисты продукта проходят обучение и демонстрацию, самостоятельная работа с агентами не требуется.

Пилот — уточнять расчёты и проверять достижение цели

В реальном потоке команда уточняет процедуру вычисления метрик: полноту первичных сведений, связи событий, границы работ и особенно **оценку объёма задач в идеальных часах**. По завершённым задачам становятся видны пропущенные работы, неточные предположения и различия сложности. ИИ-инженер совершенствует оценивание и расчёты, РП проверяет оценочные значения. Изменения методов, входных оценок и их версий сохраняются вместе с основаниями.

Если новый способ оценки объёма меняет значение показателя, это само по себе не означает улучшения или ухудшения работы. Для сравнения пересчитывают исходную базу и наблюдаемые результаты по сопоставимым правилам; если пересчёт невозможен, выделяют периоды с различными условиями и явно ограничивают вывод.

При необходимости корректируют и целевые значения. Основание — уточнённая исходная база, более точная оценка объёма, обнаруженные ограничения или согласованное изменение области. РП готовит и сохраняет основание пересмотра, прежний и новый ориентиры, версию правил, период применения и решение участников, согласовавших условия. Изменение самой содержательной цели согласуется со спонсором. Ориентир не подгоняют под уже полученное число без обоснования.

Результат — проверенные показатели и актуальные согласованные условия оценки Пилота с историей уточнений. Специалисты продукта работают с агентами непосредственно; инженер сопровождает поток. Недельные разборы и небольшие циклы PDCA ниже позволяют одновременно улучшать способ работы и достоверность его измерения.

3. Ритм работы и взаимодействие участников

РП организует регулярный разбор и закрепление решений в общем плане. ИИ-инженер готовит технический анализ и предложения. Владелец продукта и профильные специалисты подтверждают влияние на продукт и проверки; DevOps подключается к ограничениям инфраструктуры. У

выбранной работы один конкретный ответственный, назначенный в плане, а не общий список ролей. При совмещении ролей сохраняется запись результата, передавать сведения самому себе не требуется.

Когда	Что рассматривают и кто подключается	Чем заканчивается разбор
После результата задачи, MR или существенного затруднения	ИИ-инженер с помощью агента разбирает результат и участие людей; при необходимости привлекает специалиста продукта	Исправление сейчас либо сохранённый факт с гипотезой для дальнейшего анализа
Ежедневно	Агент собирает сведения; РП и участники разбирают зависшие обращения, проверки, среды и другие препятствия	Действия по устранению препятствий, ответственные и сроки; вопросы без решения остаются видимыми
Еженедельно	РП и ИИ-инженер рассматривают показатели, группы фактов и прошлые изменения; по затронутой причине подключаются специалисты продукта или DevOps	Одна-две приоритетные группы на следующую неделю, выбранные изменения и решения о проверенных изменениях
В контрольной точке Пилота	РП, ИИ-инженер и владелец продукта рассматривают подтверждённые данные, показатели и обязательные условия	Фиксация успешного завершения либо подготовленное решение о дальнейших действиях при отклонениях

Источники — задачи, MR, запуски агентов, обращения и ответы людей, результаты CI, тестовых сред и инфраструктурных проверок. Из первичных сведений формируют проверяемые наборы данных; оценочные величины подтверждает РП, метрики вычисляются по сохранённым правилам. Отсутствие данных не считается нулевыми затратами. Общие затраты на развитие агентов учитываются отдельно без двойного счёта.

Факты и выбранные работы связывают с исходной задачей/MR и сохраняют в рабочих системах. Журнал улучшений фиксирует изменения и проверки, недельная сводка — показатели, выводы и решения; повторно переписывать подробные логи не нужно. Бэклог фактов может быть списком наблюдений в этих системах: отдельный новый документ или задача на каждое наблюдение не требуются.

Анализ причин потерь, выбор меры и проверка эффекта выполняются в этом общем цикле. Подготовка и вовлечение участников реализует выбранные меры по людям: обучение, помощь, время и замещения. Встраивание агентов в процессы разработки реализует изменения маршрута: события, переходы, точки контроля и передачу работы. Результаты возвращаются в ту же недельную сводку; независимые разборы одних и тех же метрик не вводятся.

4. Как выбрать изменение и проверить решение

Сначала находят существенную потерю, затем выбирают воздействие на её причину. Например, долгое ожидание MR может требовать доступности проверяющего и замещения, а не доработки навыка агента. Недостаток данных теста может требовать сброса среды, а не новых подсказок модели. Быстрый первый ответ не устраняет очередь не выполненных ревью.

1. **Зафиксировать наблюдение.** Указать источник, условия, ожидаемое и фактическое поведение, последствия и человеческие затраты. Отделить проверенные сведения от предполагаемой

причины.

2. **Выбрать дальнейшее действие.** Конкретный локальный дефект исправить сейчас. Неясные или противоречивые случаи сохранить для анализа. Нарушение обязательного ограничения сразу ограничить или остановить, не дожидаясь недельной статистики.
3. **Определить предмет изменения.** Требования и контекст, инструкции и навыки, инструменты и тестовые среды, инфраструктура, организация HITL или границы поручаемой агентам работы. Каждая проблема не обязана превращаться в доработку агента.
4. **Согласовать ограниченную проверку.** В плане записать изменение, одного ответственного, ожидаемый эффект, сопоставимые задачи или сценарии, показатели баланса и срок разбора. РП организует согласование приоритета и времени; решения о продукте и технических ограничениях остаются у уполномоченных участников.
5. **Выполнить и оценить.** Сохранить версии и условия, проверить проблемный сценарий и следующие подходящие задачи. Сравнить время, активный труд людей и расход ресурсов; убедиться, что потери не переместились на другой этап.
6. **Закрепить, доработать или отменить.** Сохранить вывод и подтверждения. При недостатке наблюдений определить конкретный вопрос следующей проверки, а не считать изменение ни доказанным, ни бесполезным.

Для разобранной группы проблем выбирают статус: обязательное исправление, улучшение по ожидаемой отдаче, сохраняемый HITL либо работа вне текущего охвата. Сохранение участия человека — явное решение с учтёнными затратами; повторяющаяся дорогая ручная переделка требует улучшения, сужения области агентов либо признания недостаточного эффекта. Изменение режима контроля проходит предусмотренное согласование и проверку, а не происходит вследствие одного удачного прогона.

5. Конкретные циклы PLAN → DO → CHECK → ACT

Ниже — способы разобрать выбранную проблему. Они применяются в общем ритме раздела 3; отдельные встречи по каждому циклу не нужны.

5.1. Цикл задачи: поставить → получить результат → разобрать → исправить или накопить факты

Запуск: агент представил результат, создал MR, запросил помощь либо остановился. Анализ начинают и до завершения всей задачи; разбор MR не означает её приёмки.

PLAN. Уточнить с агентом понимание задачи, ожидаемый результат и проверки. Выбрать условия выполнения и зафиксировать используемые версии агентов, чтобы результаты можно было объяснить.

DO. Агент выполняет работу, человек проводит предусмотренную проверку. Замечания возвращаются агенту для исправления; фиксируются результаты, затруднения и участие людей.

CHECK. После результата или MR ИИ-инженер с помощью агента разбирает: что получилось, где потребовалась помощь, что переделывали и почему. Различает ошибку конкретного изменения и недостаток общего способа работы. Например, замечание к одному MR ещё не доказывает, что нужно менять общий навык.

ACT. Для каждого существенного наблюдения выбирают один из двух путей:

- **Исправить сейчас.** Конкретную ошибку вернуть агенту; понятный локальный недостаток инструкции, контекста или инструмента исправить и проверить на затронутом сценарии. Изменение общего навыка требует проверки побочных последствий, а не только успешного повтора одной задачи.
- **Сохранить в бэклог фактов.** Если причина неясна, поведение противоречиво или исправление требует анализа нескольких случаев, сохранить наблюдение и связать его с похожими. Текущую задачу довести до результата с доступной помощью человека, не ожидая общего решения.

В бэклоге фактов достаточно ссылки на задачу/MR/запуск, ожидаемого и фактического поведения, условий и версии, последствий и человеческих затрат. Предполагаемую причину отмечают как гипотезу. Такой факт ещё не является поручением «дописать навык»: способ исправления предстоит выбрать. При риске нарушения обязательных ограничений сразу останавливают затронутую работу или ограничивают её, даже если причина пока не установлена.

Результат цикла: исправленная задача и проверенное локальное улучшение либо задача, завершённая при помощи человека, и сохранённые факты для следующего анализа. Передача задачи человеку и его затраты отражаются явно.

Подробнее: работа ИИ-инженера, план мониторинга и проверка измерений.

5.2. Недельный цикл: сгруппировать факты → проверить причины → устранить выбранные группы

Запуск: недельный разбор состоявшихся задач, накопленных фактов и показателей. Новые сложности не отменяют разбор и не обнуляют уже полученные результаты.

PLAN. Сопоставить случаи и объединить их в небольшое число групп по предполагаемой общей причине: требования и контекст; навыки; инструменты и среды; организация взаимодействия. Не создавать отдельную группу под каждую задачу. Выбрать на следующую неделю одну-две наиболее существенные группы, ожидаемый эффект и проверки. Приоритет определяют последствия, человеческие трудозатраты и стоимость устранения, а не только число повторений.

DO. ИИ-инженер анализирует выбранные группы и меняет нужные инструкции, навыки, контекст или инструменты; изменения процессов и сред выполняет с ответственными участниками. Сначала проверяет проблемные случаи, затем применяет изменение на ограниченном наборе подходящих задач. Версии и условия сохраняются.

CHECK. На следующем разборе проверить: причина устранена или ошибка лишь переместилась; уменьшились ли труд людей, длительность и расход ресурсов; сохранилось ли качество. Противоречивые результаты разделить по условиям, а не усреднять: например, улучшение одной модели может ухудшить работу другой. Проверки на старых сценариях подтверждают исправление этих случаев, но сами по себе не доказывают эффект на всём рабочем потоке.

ACT. Закрепить подтверждённое улучшение, отменить неудачное либо назначить следующую ограниченную проверку с конкретным вопросом и сроком. Для каждой разобранной группы определить дальнейший статус: обязательное исправление; улучшение по ожидаемой отдаче; сохраняемый HITL; работа вне текущего охвата. Остальные факты остаются в очереди и не блокируют весь Пилот.

Редкий предметный вопрос может оставаться человеку. Повторяющаяся дорогая ручная переделка требует отдельного решения: улучшить автоматизацию, сузить область агентов либо признать недостаточный эффект. Сохраняемый HITL должен быть предусмотрен процессом и учтён в затратах, а не обозначен как «потом как-нибудь исправим».

Подробнее: ритм разбора, улучшение агентов и процесса.

5.3. Выполнение работы агентов: токены, стоимость и длительность шагов

Запуск: завершён шаг или запуск агента, доступны сведения о времени и токенах; либо во время выполнения обнаружены повторные попытки или превышение согласованного лимита. Случай разбирают в цикле задачи, накопленные данные — еженедельно.

PLAN. По разбору задачи или недельной сводке выбрать дорогой либо долгий шаг: повторное чтение контекста, лишние попытки, избыточную декомпозицию или генерацию. Определить ожидаемое сокращение без потери качества.

DO. Проверить одно выбранное изменение: состав контекста, навык, инструмент, модель или организацию шагов.

CHECK. Сравнить токены, стоимость и время выполнения на сопоставимых задачах. Токены и длительность — разные величины; стоимость учитывает модель и кэш. Проверить, что экономия не оплачена ростом ручных исправлений.

ACT. Закрепить улучшение в рабочей версии либо отменить его. Неподтверждённую причину вернуть в бэклог фактов с условиями следующей проверки.

5.4. Ожидание агентов: очереди, инфраструктура и внешние зависимости

Запуск: агент остановился в ожидании ресурса или внешнего результата либо такое ожидание обнаружено при ежедневном просмотре текущей работы. Разбор начинают во время простоя; после его окончания фиксируют полную длительность, а повторяющиеся причины рассматривают еженедельно.

PLAN. Найти интервал простоя и его источник: ожидание runner, среды, квоты, результата внешней системы. Ожидание человека разбирается отдельно, в цикле ожидания человека.

DO. Устранить выбранную причину: обеспечить ресурс, изменить параллелизм, подготовку среды или порядок зависимых шагов.

CHECK. Проверить длительность ожидания и общий срок задачи, а также изменение инфраструктурных расходов. Перенос очереди на другой этап не считается устранением.

ACT. Сохранить настройку или порядок работы, если выигрыш оправдан; иначе пересмотреть решение либо зафиксировать внешнее ограничение.

5.5. Работа человека: человеко-часы проверки и помощи

Запуск: завершено ревью, уточнение, проверка или другое вмешательство человека и доступны сведения о затратах. Существенную ручную переделку разбирают сразу; суммарные затраты на сопоставимый объём — еженедельно.

PLAN. Выбрать вмешательство, которое требует существенного труда: восстановление контекста, длинное ревью, повторное объяснение или ручная переделка. Сформулировать, что агент должен подготовить лучше.

DO. Изменить состав материалов, предварительные проверки или навык агента. Обязательное предметное решение человека сохранить; не заменять улучшение формальным снятием проверки.

CHECK. Сравнить активные человеко-часы на сопоставимый принятый объём, включая повторные проверки и помощь инженера. Убедиться, что работа не переместилась в HOTL, тестирование или поздние доделки и качество не ухудшилось.

ACT. Закрепить более дешёвое взаимодействие либо продолжить поиск причины. Цель — существенно уменьшить труд людей, а не число обращений или довести HITL до нуля.

5.6. Ожидание человека: от обращения до начала работы

Запуск: обращение осталось без ответа в течение двух часов рабочего времени либо при ежедневном разборе обнаружена ожидающая проверка, в том числе уже подтверждённая первым ответом. После начала содержательной проверки фиксируют время ожидания; повторяющиеся задержки разбирают еженедельно.

PLAN. По событиям найти зависшие обращения и причину: запрос не замечен, проверяющий недоступен, не назначен заместитель или непонятно требуемое действие.

DO. Исправить адресацию и уведомления, назначение проверяющего, замещение или содержание запроса. Обеспечить ответ в течение двух часов рабочего времени; все вопросы без решения разбирать на ежедневной встрече.

CHECK. Измерить ожидание до начала содержательной проверки отдельно от времени первого ответа и длительности самой проверки. Проверить общий срок задачи и нагрузку участников: быстрый ответ не должен скрывать очередь ещё не выполненных ревью.

ACT. Закрепить работающий порядок либо изменить доступность и распределение проверок. Дополнительное время людей учитывать как затраты, даже если ожидание сократилось.

6. Когда заканчивается Пилот и что сохраняется дальше

Пилот завершается по согласованным показателям и обязательным условиям, а не по отсутствию новых проблем. При их выполнении сохраняемый HITL и очередь необязательных улучшений не мешают успешному завершению. РП фиксирует успешное завершение без отдельного решения спонсора о том, принимать ли достигнутые согласованные показатели.

Если условия не выполнены, РП сохраняет отклонения и представляет спонсору предложение: продолжить Пилот, изменить условия с продолжением либо остановить. Обоснованные уточнения методики и целевых значений в ходе Пилота выполняются по этапному порядку; при итоговой проверке видны актуальные условия, основания изменений и сопоставимые расчёты. Сам по себе пересмотр без проверки результатов не закрывает Пилот. Подробный порядок — в чек-листе завершения Пилота.

Малое число задач не отменяет недельный цикл: уточняют гипотезу и следующие проверки, сохраняя ограничения выводов. Процесс не заканчивается достижением нулевого HITL или пустого бэклога. После Пилота остаются закреплённые версии и правила, известные границы работы агентов, предусмотренное участие людей, ответственные за сопровождение и очередь дальнейших улучшений.

Связанные материалы: расчёты метрик, ежедневный сбор сведений, работы Пилота по улучшению и проверке результатов.

Проба изменения контроля в PIL-13 — специализированный цикл этого процесса: гипотеза, ограниченная область, измерения и решение. Изменённые правила и работоспособность обмена подтверждаются в PIL-10/12. В PIL-17 та же проба повторно не заводится; запись улучшения и решение используются совместно.

Работы процесса и подробные действия ролей

Основная принадлежность работы определяется её результатом и записана в чек-листе. Ниже показаны работы этого процесса; ссылки на другие процессы описывают взаимодействие, без повторного назначения той же работы. ID и предшественники находятся в исходной работе.

PoC.

- POC-18. Адаптировать агентов на задачах PoC — ответственный ИИ-инженер.
- POC-19. Свести результаты проверок PoC — ответственный ИИ-инженер.
- POC-20. Подготовить и проверить рабочую версию агентов — ответственный ИИ-инженер.
- POC-21. Согласовать метрики и правила мониторинга — ответственный РП.
- POC-22. Рассчитать и согласовать с РП оценочные значения для сравнения — ответственный ИИ-инженер.
- POC-23. Настроить и проверить ежедневный сбор сведений — ответственный ИИ-инженер.
- POC-24. Проверить пробную недельную сводку — ответственный РП.

Пилот.

- PIL-14. Ежедневно собирать сведения и вычислять метрики — ответственный ИИ-инженер.
- PIL-15. Согласовывать оценочные значения с РП — ответственный ИИ-инженер.
- PIL-16. Проверять недельные сводки и полноту данных — ответственный РП.
- PIL-17. Улучшать агентов и способ работы по выявленным ограничениям — ответственный ИИ-инженер.
- PIL-13. Проверять изменения режима человеческого контроля — ответственный РП.
- PIL-18. Закрепить проверенные версии и возможность восстановления — ответственный ИИ-инженер.
- PIL-19. Подвести количественные итоги по целям внедрения — ответственный РП.
- PIL-20. Проверить качество, нагрузку людей и стоимость — ответственный РП.
- PIL-21. Описать проверенный способ работы и границы его применения — ответственный ИИ-инженер.
- PIL-22. Устранить блокирующие отклонения и подтвердить результат — ответственный РП.

Подготовка и вовлечение участников

Процесс обеспечивает людей, способных и готовых выполнять новые обязанности: выделенное время, обучение, освоение действий своей роли и доступную помощь. Назначение в паспорте должно стать реальным участием, а участие команды внедрения — помогать освоению работы специалистами продукта.

1. Границы, участники и результат

РП организует назначения, загрузку и обучение, владелец продукта обеспечивает участие специалистов, ИИ-инженер демонстрирует работу и сопровождает освоение. При совмещении ролей сведения и решения всё равно сохраняются в рабочих документах.

Результат: для нужных действий назначены конкретные участники и заместители, выделено время; обучение подтверждено материалами и демонстрацией, на Пилоте — фактами выполнения обязанностей. Дальнейшая помощь и сопровождение обеспечены. Назначения и ресурс находятся в паспорте проекта, подтверждения — в материалах обучения и обычных рабочих системах.

Сами события, необходимые роли, проверки и условия продолжения определяются в процессе встраивания агентов. Здесь обеспечивают исполнителей этих действий. Например, наличие точки ревью и требование вердикта относятся к маршруту; назначенный разработчик, его время и умение дать замечания — к подготовке участников. Качество регламента не доказывает освоения работы человеком.

2. Подготовка: назначить, выделить время, познакомить

Для начала собирают деятельную команду с РП, ИИ-инженером и нужным предметным специалистом. В паспорте записывают исполнителей, выделенное время, замещения и доступную помощь. Владелец продукта подтверждает, что участие включено в рабочую загрузку. При внешней помощи на стороне продукта сохраняется участник, обеспечивающий действия своих коллег.

Назначения PREP-01 и готовность участия в Пилоте POC-02 относятся к этому процессу. PREP-05 уточняет доступность и замещения и организует коммуникации, не создавая второй состав команды.

РП организует рабочий чат, ежедневную встречу, еженедельный доклад Спонсору и содержательную реакцию в течение двух часов рабочего времени. Эти договорённости обеспечивают доступность людей. Технические каналы и доступ агентов к сообщениям настраиваются в рамках технической готовности, а события обращений описываются в ЖЦ задач и Git.

ИИ-инженер проводит вводный семинар: показывает поручение задачи, вопрос агента, предъявление результата и ответ человека. Участники выполняют предусмотренное учебное упражнение, разбирают действия своих ролей. Сохраняются материалы, состав и подтверждения. Это знакомство с процессом, а не требование самостоятельно вести продуктовые задачи на PoC.

Выход: назначения и время записаны в паспорте, встречи заведены, вводный семинар проведён и подтверждён.

3. PoC: обучить на работе инженера

ИИ-инженер ведёт взаимодействие с агентами и показывает специалистам работу на их продукте. Участникам доводят новый процесс, демонстрируют постановку и уточнение задачи, ревью, замечания и повторную проверку. Они предоставляют предметные сведения и решения, разбирают свои будущие действия и затруднения. Отдельное занятие не обязательно, если демонстрация проведена в обычной работе PoC и сохранены подтверждения.

На поздних задачах специалист может подключаться через инженера: тот проверяет материалы перед предъявлением коллеге и обратную связь перед возвратом агенту. Так участник знакомится с новым способом работы без необходимости самостоятельно разбирать сырые материалы. Устройство этого обмена описывается в общем маршруте.

Перед Пилотом РП подтверждает назначения, время, замещения, обучение и доступную помощь. Если целевые инструменты или действия отличаются от показанных, участникам представляют эти различия. Личные доступы проверяются в инфраструктурных работах; здесь используют их подтверждения.

Выход: есть материалы обучения, состав участников, демонстрация и разобранные вопросы; участие на Пилоте обеспечено. Самостоятельная работа специалистов на PoC не является обязательным гейтом.

4. Пилот: освоить обязанности и закрепить сопровождение

Специалисты продукта непосредственно выполняют действия своих ролей. РП проверяет по фактам работы, кто уже действует самостоятельно, кому нужна помощь, у кого не хватает времени или предметных знаний. При затруднениях ИИ-инженер помогает выполнить действие и освоить способ работы; помощь и ручное выполнение вместо участника различаются в данных.

Работа PIL-05 подтверждает освоение обязанностей участниками. Для этого достаточно ссылок на реальные поручения, ответы, ревью и проверки; новый учебный журнал и искусственные задания не вводятся. Выполнение всех действий инженером внедрения не подтверждает освоение командой. Работоспособность самого маршрута отдельно подтверждается в PIL-10.

Новые участники получают обучение и сопровождение по тем же правилам. Для инженерных обязанностей используют реальные улучшения, если они понадобились. В журнале улучшений указаны фактический исполнитель и команда; при работе инженера внедрения — почему её не выполнил инженер продукта. Ненужное улучшение ради учебной отметки не создают.

К завершению Пилота в паспорте закреплены поддержка, время, доступы и замещения. Полная передача инженерного сопровождения не обязательна: согласованная внешняя помощь может сохраняться с учётом её ресурса и стоимости.

5. Взаимодействие с другими процессами

На ежедневной встрече устраняют отсутствие исполнителя, замещения или времени; организуют необходимую помощь. Если человек обучен и доступен, но запрос пришёл не туда или переход задачи не работает, исправление относится к встраиванию агентов. Проблемы подключения и прав передаются в техническую готовность.

Повторяющиеся потери и эффект изменений рассматриваются в общем процессе эффективности. Здесь выполняют выбранные меры по людям: меняют загрузку, обучают, назначают помощь или замещение. Отдельный параллельный разбор тех же метрик не создаётся. Неблокирующий чек-лист условий команды помогает выбрать направления подготовки.

Работы процесса и подробные действия ролей

Основная принадлежность работы определяется её результатом и записана в чек-листе. Ниже показаны работы этого процесса; ссылки на другие процессы описывают взаимодействие, без повторного назначения той же работы. ID и предшественники находятся в исходной работе.

Подготовка.

- PREP-01. Собрать команду внедрения — ответственный РП.
- PREP-05. Организовать участие и коммуникации — ответственный РП.
- PREP-06. Провести вводный семинар для участников PoC — ответственный ИИ-инженер.

PoC.

- POC-02. Подтвердить состав команды и время на Пилот — ответственный РП.
- POC-06. Обучить команду и продемонстрировать работу с агентами — ответственный РП.

Пилот.

- PIL-05. Обеспечить освоение обязанностей участниками Пилота — ответственный РП.
- PIL-06. Закрепить дальнейшее сопровождение и развитие агентов — ответственный РП.

Обеспечение технической готовности

Процесс обеспечивает рабочий путь от поручения задачи до проверенного изменения: инференс, ИИ-платформа, конвейер, среды и нужные права. Готовность подтверждается выполненной операцией из требуемого контура; установленный сервис, открытый порт или выданный аккаунт сами по себе недостаточны.

1. Входы и распределение действий

Входы — область и типы работ из паспорта, корпоративные ограничения, существующие системы и доступный ресурс. ИИ-инженер задаёт требования платформы и агентов, DevOps обеспечивает назначенные инфраструктурные действия. Разработчик и тестировщик подтверждают сборку, данные и сценарии. РП организует согласования и личные проверки доступа.

Требования и подтверждения ведутся в одной карточке инфраструктуры: инференс, платформа, инструменты конвейера, тестовые среды, размещение, связи и права людей, систем и платформы. Не создаются отдельные несвязанные реестры доступа и Git.

2. Подготовка: описать, обеспечить, проверить

1. По выбранной области определить конфигурацию и необходимые операции. Отделить существующее от устанавливаемого, требования от фактических проверок. Для on-prem выделение ресурсов и доступов предшествует установке; участие внедрения не даёт прав на системы предприятия.
2. Обеспечить платформу, инференс и подключения к Git, трекеру, чату, CI и материалам. Проверить не только вход, но и разрешённое действие, обработку ошибки и доступность диагностики.
3. Провести существующую ревизию через сборку, хранилище артефактов и деплой. Зафиксировать SHA/digest и фактически запущенную версию. Это проверка инфраструктуры, а не доказательство способности агента решить задачу продукта.
4. Подготовить среды по нужному параллелизму, данные, сброс и применимые механики: моки, время, эмуляторы. Нужен минимум e2e выбранных сценариев, похожий на production по существенным свойствам. Полный набор инструментов не требуется до первого ограниченного шага, но необходимый инструмент обеспечивается до зависимой задачи.
5. ИИ-инженер проверяет системы и связность, а РП организует личные действия пользователей. У каждого вида проверки своё подтверждение.

Выход: карточка и выполненные технические проверки для начала PoC. Неполная готовность будущего расширения отражается как ограничение области, а не как успешная проверка.

3. PoC: уточнить целевой контур Пилота

PoC может использовать временные Git, трекер и среды. По мере проб выявляются недостающие операции и уточняются требования. Перед Пилотом карточка сопоставляет временную и целевую конфигурации: что изменилось, какие аккаунты, связи, среды и проверки нужны повторно.

DevOps обеспечивает целевые ресурсы и доступы; разработчик — сборку и совместимость ревизий; тестировщик — пригодность данных и e2e; ИИ-инженер проверяет из целевого рабочего контура. Каждый участник лично подтверждает свои операции. Старое успешное обращение из временной среды не считается доказательством нового подключения.

К гейту Пилота проверены необходимые системы, целевой маршрут и источники мониторинга. Неожиданная ручная операция либо автоматизируется, либо явно включается в порядок работы с человеком и учитывается как труд и ожидание. Ограничение не прячется в инструкции инженера.

4. Пилот: поддерживать и уточнять требования

В рабочем потоке различаются очередь запуска, зависимость, ожидание среды и отказ. По плану мониторинга видны длительности, открытые ожидания, нагрузка и затраты. Устранение одной очереди не считается улучшением, если выросли труд людей, расходы или ошибки.

При сбое сохраняется затронутая операция и ревизия; после восстановления она проверяется повторно. Общий инцидент не приписывается всем задачам целиком. Изменения лимитов, моделей, ресурсов, сред и связей документируются с периодом применения; расчёты соотносятся с этой конфигурацией.

К завершению Пилота карточка отражает итоговые требования и фактические изменения, а поддержка имеет исполнителей и ресурс. Нужно показать, на какой конфигурации получены показатели, а не просто утверждать, что инфраструктура выдержала нагрузку.

5. Подтверждения, ритм и связи

После изменения выполняется зависимая проверка; ежедневно разбираются недоступность и очереди; еженедельно выбираются приоритетные улучшения. В журнале улучшений сохраняются реальные исполнители, конфигурации и эффект. Корпоративные решения запрашиваются через уровень предприятия, а не подменяются дополнительной ролью в проекте.

Конкретные механики и пример — в тестовой инфраструктуре; правила кода и CI — в Git-карточке. Подробные команды и проверки определяет гайд DevOps и инструкция выбранной платформы.

Проверка чтения конкретных выбранных задач в PREP-17 дополняет общие проверки подключений: подтверждает доступ к описаниям и вложениям из контура агента. Проверка понимания требований выполняется позже на PoC; чтение не означает достаточности постановки.

Работы процесса и подробные действия ролей

Основная принадлежность работы определяется её результатом и записана в чек-листе. Ниже показаны работы этого процесса; ссылки на другие процессы описывают взаимодействие, без повторного назначения той же работы. ID и предшественники находятся в исходной работе.

Подготовка.

- PREP-07. Оформить и согласовать карточку инфраструктуры — ответственный ИИ-инженер.
- PREP-08. Развернуть или подключить платформу — ответственный DevOps.
- PREP-09. Подключить и проверить инференс — ответственный ИИ-инженер.
- PREP-10. Организовать хранение материалов внедрения — ответственный ИИ-инженер.
- PREP-11. Настроить связи и доступы системам и ИИ-платформе — ответственный DevOps.

- PREP-12. Подготовить рабочие места и доступы людям — ответственный DevOps.
- PREP-13. Проверить технический путь сборки и развёртывания — ответственный DevOps.
- PREP-14. Подготовить тестовые среды продукта — ответственный DevOps.
- PREP-27. Подтвердить готовность систем и связности — ответственный ИИ-инженер.
- PREP-30. Подтвердить доступы участников команды — ответственный РП.
- PREP-17. Проверить чтение выбранных задач агентами — ответственный ИИ-инженер.

PoC.

- POC-07. Уточнить инфраструктуру и доступы для Пилота — ответственный DevOps.
- POC-08. Подготовить системы и ресурсы Пилота — ответственный DevOps.
- POC-09. Настроить связность и выдать доступы — ответственный DevOps.
- POC-10. Подготовить тестовые среды для потока Пилота — ответственный DevOps.
- POC-11. Проверить техническую готовность из контура Пилота — ответственный ИИ-инженер.
- POC-12. Организовать личную проверку доступов участников — ответственный РП.

Пилот.

- PIL-07. Поддерживать рабочую инфраструктуру и доступы — ответственный DevOps.
- PIL-08. Обеспечивать воспроизводимость тестирования — ответственный DevOps.
- PIL-09. Актуализировать требования к инфраструктуре и инференсу по результатам Пилота — ответственный DevOps.

НА ПРЕДПРИЯТИИ

Гайд роли: Спонсор

Инициирует внедрение и принимает решения о переходах, ресурсах и изменениях условий в пределах полномочий предприятия. Не готовит вместо команды предложения по трансформации и не пересматривает достигнутый успех произвольно.

Назначение и распределение работ

Назначается со стороны предприятия. Команда внедрения готовит анализ, варианты и материалы; Спонсор рассматривает и утверждает либо отклоняет предложения. Участие консультантов не заменяет управленческую ответственность предприятия.

Конкретные участники, принадлежность к команде и выделенное время — в паспорте проекта. У каждой работы один ответственный; соисполнитель подтверждает свой вклад и не наследует все полномочия ответственного. ID и непосредственные зависимости берутся из чек-листов; порядок чтения ниже не создаёт новых зависимостей. Существующий подходящий результат проверяется и используется по ссылке.

Как выполнять работу роли

На встрече требовать разницу между состоянием сейчас и предложением, основания расчёта, ограничения и варианты. Решение должно иметь область, условия и дальнейшее действие. Достигнутые согласованные показатели Пилота и обязательные условия означают успешное завершение, которое фиксирует РП без повторного одобрения Спонсором.

Результат сохраняется в предусмотренном рабочем документе или событии системы. Отдельный отчёт каждого участника по каждой работе не требуется. На PoC регулярное взаимодействие с агентами ведёт ИИ-инженер, специалисты обучаются и подключаются через него; на Пилоте работают непосредственно.

На уровне предприятия

Прогнозирование эффектов. Инициировать этап и обеспечить участие команды внедрения и предметных специалистов. В ходе проб рассматривать промежуточные наблюдения, ограничения и возникающие развилки. К итоговой встрече команда готовит предложение о трансформации: текущая и предлагаемая оргструктура и стоимость, ожидаемые параметры поставки, бюджет, сроки и новые риски. На встрече по презентации принять решение о программе, продолжении ограниченных проб либо отказе. Офис и отдельная команда прогнозирования для старта не требуются.

Пилотная волна. Утвердить предложенную область программы, ответственного руководителя и доступный ресурс. Рассматривать препятствия, которые не могут решить РП и владельцы продуктов. Успех продукта оценивать по его согласованным условиям, а не по внешнему впечатлению от презентации.

Решения по следующим волнам. Рассматривать готовность условий и предложения руководителя программы, учитывая разные продукты и общие расходы. Успешный Пилот одного продукта не разрешает подключение остальных автоматически. Подробные работы волн здесь не планируются.

К подведению итогов. Должны быть доступны подтверждённые результаты, расходы, ограничения, невыполненные обязательства и предложение дальнейшего периода. Это требования к подготовленным материалам, не отдельный план финального этапа.

Подробности совместных действий — в процессе целей и приоритетов и управлении программой.

Подготовка

PREP-02. Оформить цели и границы внедрения

Спонсор — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Рассмотреть предложение РП и владельца продукта о цели, границах и ограничениях внедрения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Утверждено направление изменений, а не придуманные на Подготовке численные показатели.

PREP-29. Принять решение о переходе к PoC

Спонсор — ответственный. Ответственный по работе — Спонсор. Работа, результат и предшественники.

1. На встрече рассмотреть подтверждения гейта и принять решение о переходе к PoC либо дальнейших действиях.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Решение, условия и ограничения записаны; пустые формы не считаются готовностью.

PoC

POC-05. Принять решение о переходе к Пилоту

Спонсор — ответственный. Ответственный по работе — Спонсор. Работа, результат и предшественники.

1. Рассмотреть результаты PoC, условия Пилота и обязательства участников; принять решение на встрече.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Разрешение опирается на проверенный поток и ресурс, а не успешную демонстрацию одной задачи.

Пилот

PIL-01. Уточнить критерии и период итоговой оценки

Спонсор — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Рассмотреть обоснованные предложения об изменении целей и условий успешности по принятому порядку.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Прежние условия, новые ориентиры и основания изменения сохранены.

PIL-04. Определить дальнейшие действия при невыполнении условий Пилота

Спонсор — ответственный. Ответственный по работе — Спонсор. Работа, результат и предшественники.

1. При невыполнении условий выбрать продолжение, изменение условий с продолжением или остановку.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Успех не объявляется только по желанию продолжать; выполненные согласованные условия не требуют повторного принятия Спонсором.

Связь с процессами и источниками

Порядок совместных действий раскрывают гайды управления внедрением, подготовки команды, технической готовности, работы с агентами и управления эффективностью. Показатели и вычисления задаёт план мониторинга; выполненные улучшения — журнал улучшений.

Гайд руководителя программы

Руководитель программы обеспечивает исполнение общих решений предприятия, координацию продуктов, ресурсов и зависимостей. Он готовит предложения Спонсору и организует работу после решений, не подменяя владельцев продуктов, РП и технических исполнителей.

Назначение и распределение работ

Назначается со стороны предприятия с полномочиями координировать продукты и общие функции. Специалисты внедрения могут готовить анализ, материалы, технические решения и сводки; ответственность за программу и межпродуктовые приоритеты остаётся у назначенного руководителя.

Состав, время и полномочия фиксируются в паспорте программы. Совмещение с РП продукта допустимо, но время и расходы не дублируются. Конкретные работы назначаются в трекере с одним ответственным; частное поручение не даёт права изменить цель всей программы.

Порядок совместных действий — в управлении программой. Ниже раскрыта работа роли на имеющихся этапах, без новых ID и гейтов предприятия.

Прогнозирование эффектов

Организовать ограниченные пробы

До формального запуска программы координацию выполняет инициатор или назначенный участник команды внедрения; отдельная команда прогнозирования не создаётся.

1. Уточнить вопрос решения, доступный ресурс и продукты для проб. Использовать критерии выбора: достаточный бэклог разработки, доступные специалисты и время на исправления.
2. Назначить участников и способ решения возникающих вопросов. Обеспечить существующие рабочие записи; не требовать офиса и полного паспорта программы до начала проб.
3. В ходе этапа показывать Спонсору существенные ограничения и варианты, не скрывая неудачные случаи до финальной презентации.

Результат роли: ограниченные пробы выполняются, вопросы имеют адресатов, материалы отражают действительный опыт и допущения.

Подготовить решение о программе

1. С командой внедрения собрать предложение: текущая и предлагаемая оргструктура и стоимость, параметры поставки, ресурс, бюджет, сроки и риски. Отличить прогноз от подтверждённого результата.
2. Подготовить итоговую встречу с вариантами: запуск, продолжение проб либо отказ. Не передавать Спонсору сырые логи вместо обоснованного предложения.
3. Сохранить принятое на встрече решение и его условия. При запуске оформить программу, не копируя заново подходящие результаты проб.

Результат роли: есть решение и исполнимые дальнейшие обязательства, а не документ, оставленный без контрольной точки.

Пилотная волна

Закрепить область, участников и ресурс

1. С владельцами продуктов согласовать состав первой волны, порядок подключений и необходимые общие работы. Зафиксировать это в паспорте программы.
2. Проверить выделенное время РП, инженеров и общих специалистов. Общая занятость не должна превышать доступность только потому, что каждый продукт посчитал один и тот же резерв своим.
3. Указать полномочия решений, существующие форматы сверок, очередь вопросов и место записей. Участие производственных специалистов обеспечивает их РП и владелец продукта.

Результат роли: локальные паспорта и общие условия совместимы, нет работы без ответственного ресурса. Подробности выбора — в процессе состава и волн.

Разрешать общие зависимости

1. На сверке выделять вопросы, которые продукт не может решить локально: ресурс инфраструктуры, доступ, занятость общей команды, корпоративное ограничение.
2. Получать варианты у назначенных специалистов, выбирать в пределах своих полномочий либо выносить Спонсору с последствиями по стоимости, срокам и области.
3. Назначать следующее действие и проверять полученный результат. Отправленное письмо и обещание не закрывают зависимость.

Результат роли: командам доступны условия или согласованное изменение обязательств. Техническое выполнение — по процессу общей инфраструктуры.

Проверять продвижение и качество общей картины

1. Получать сведения об этапах и результатах проектов с версиями правил и ограничениями, а не только общий процент «внедрения».
2. Вместе с методологом различать завершённый PoC, успешный Пилот, производственный эффект и затраты. Несопоставимое показывать раздельно.
3. Выбирать общие приоритеты по данным и проверять исполнение прежних решений. Локальный успех при выполнении условий фиксирует РП, повторное субъективное принятие его программой не вводится.

Результат роли: решения опираются на сопоставимые результаты, а общие препятствия не перекладываются на отдельные команды без ресурса.

Условия следующих волн

Для предложения подключения следующих продуктов проверить доступность людей, технических условий, общих требований и опыта с названными ограничениями. Согласовать существенные изменения состава, сроков и ресурса по полномочиям и сохранить основания. Успех первого продукта не распространяется автоматически на все остальные.

Этот гайд не создаёт подробного перечня работ Масштабирования. При сопровождении уже утверждённых волн применяется тот же порядок ресурсов, зависимостей и решений; проект каждого продукта сохраняет собственные гейты.

Готовность к подведению итогов

Должны быть доступны актуальная область программы, подтверждённые переходы и результаты продуктов, фактические общие расходы, история изменений целей и оставшиеся обязательства с исполнителями. Поддержка после календарного окончания обеспечена ресурсом. Эти сведения готовятся в ходе программы; подробные работы финального этапа здесь не планируются.

Рабочие источники — паспорта, трекер, журналы продуктов и записи решений. Агент помогает их собрать и объяснить; руководитель программы обеспечивает проверку и действия, не вводя второй ручной отчёт о каждой продуктовой задаче.

Гайд методолога офиса внедрения

Методолог закрепляет единые требования предприятия к агентской разработке, проверяет их применение и сопоставимость результатов. Основной предмет — допустимые модели и инференс, получение инфраструктуры, обязательные метрики, правила отчётности и исключения. Методическая помощь и обучение обеспечивают применение этих требований на продуктах.

Назначение и распределение работ

Офис — функция организации, а не роль одного человека. Методолог может выполнять эти функции без отдельного офиса в первой волне. Полномочия, время и участие записываются в паспорте программы.

Специалист внедрения может готовить предложения и материалы на основе опыта. Уполномоченные участники предприятия согласуют корпоративные требования; методолог закрепляет принятые решения и проверяет применение. Знание платформы не даёт права самостоятельно менять политики или выделять ресурсы.

На этапе прогнозирования эффектов методолог и офис не являются обязательными участниками. Первые пробы ведёт команда внедрения с предметными специалистами. Здесь работа роли начинается с условий пилотной волны.

Пилотная волна

Закрепить минимальные общие требования

1. Получить действующие политики и технические решения: допустимые модели, размещение и передачу данных, виды программного доступа, порядок получения ресурсов.
2. С профильными участниками сформулировать применимые требования к первым продуктам: область действия, проверяемое условие, подтверждение и полномочие исключения. Не требовать одинаковый стек и все инструменты сразу.
3. Согласовать предложения у участников с соответствующими полномочиями. Сохранить версию и место действующих материалов; не считать рекомендацию команды внедрения уже принятым стандартом.

Результат роли: РП и инженеры могут применить согласованные требования в паспорте, инфраструктуре и процессах, не выясняя их заново по личным разговорам. Совместный порядок — в общей инфраструктуре.

Обеспечить сопоставимость показателей

1. Определить обязательные сведения программы: состав продуктов, этап и период, определения метрик, версии конфигурации и методики, происхождение оценок, полнота и ограничения.
2. Сопоставить планы мониторинга продуктов с общими правилами. Проверить, что одинаковое название означает одинаковую величину; различия границ сохранять, а не маскировать общим заголовком.
3. На первой сводке проверить несколько значений до первичных источников, оценки — до подтверждений РП. Не объединять несопоставимое и не усреднять медианы или проценты

улучшения продуктов.

Результат роли: общая картина различает подтверждённое и ожидаемое, продвижение внедрения и эффект. Расчёты — в метриках, порядок — в сведении результатов.

Помочь применить методологию на продукте

1. С РП и инженером выявить конкретный вопрос применения: что отличается от исходного правила и как это влияет на проверки, полномочия или данные.
2. Подготовить допустимую адаптацию либо вопрос исключения. Сохранить область, причину, ограничения и решение; отсутствие данных не превращать в отмену требования.
3. Проверить применение на задаче или рабочем документе. Материалы обучения должны показывать собственный продукт и реальные действия ролей; самостоятельная работа специалистов на PoC не становится новым обязательным гейтом.

Результат роли: действует понятная редакция правила, команда может выполнить его и подтвердить результат. Подробности — в процессе методологии и опыта.

Применение в последующих волнах

Проверить переносимость практики

1. Получить описание улучшения с проблемой, версиями, проверками, человеческим участием и ограничениями. Число написанных инструкций не является результатом.
2. Сравнить условия нового продукта: архитектуру, зависимости, доступы, типы задач и контроль. Выделить необходимые проверки адаптации.
3. После проверки сохранить применимость и исключения. Разрешение снизить HITL в одной области не становится общей корпоративной отменой контроля.

Выделить общую причину и нужную помощь

1. По повторяющимся вопросам отделить общий недостаток правил или инструмента от частной ошибки продукта.
2. Подготовить ограниченное изменение и проверку его результата; руководитель программы обеспечивает назначение исполнителя и ресурс.
3. Зафиксировать выполненную работу и фактических исполнителей в существующих записях. Для инженера внедрения — почему работу не выполнил инженер продукта. Полная передача поддержки и устранение всего HITL не являются самостоятельными условиями успеха.

Эти процедуры задают способ применения принятых правил и опыта; подробный план Масштабирования и новые гейты предприятия здесь не вводятся.

Ритм работы и исключения

Участвовать в общем ритме программы, разбирать повторяющиеся расхождения сводок и конкретные запросы команд. Срочное ограничение передавать участнику с полномочиями по мере обнаружения. Для исключения нужны область, основание, срок пересмотра и согласование; до него действует разрешённая конфигурация.

Не нужно проводить отдельный комитет для каждой интерпретации методологии. Нужны версия принятого правила, доступность участникам и проверка того, что оно используется.

Готовность к подведению итогов

Должны быть доступны действующие требования, история исключений, сопоставимые результаты, общие расходы и проверенные практики с назначенной поддержкой. Неясности и непроверенные условия сохраняются явно. Это входные материалы финального этапа, а не его подробный план работ.

Организация функции и ориентиры встреч — в материале об офисе внедрения.

Гайд роли: Владелец продукта

Определяет продуктовую область внедрения, приоритеты потока и смысл принимаемого результата. Подтверждает участие специалистов и границы сравнения эффекта.

Назначение и распределение работ

Назначается со стороны предприятия с полномочиями на продуктовые приоритеты и приёмку. Специалист внедрения может подготовить анализ и варианты, но не принимает за него решения о продукте. Выделенное время и замещение записываются в паспорте.

Конкретные участники, принадлежность к команде и выделенное время — в паспорте проекта. У каждой работы один ответственный; соисполнитель подтверждает свой вклад и не наследует все полномочия ответственного. ID и непосредственные зависимости берутся из чек-листов; порядок чтения ниже не создаёт новых зависимостей. Существующий подходящий результат проверяется и используется по ссылке.

Как выполнять работу роли

Выбирать поток разработки функциональности с достаточным бэклогом и временем на несколько раундов исправлений. Различать цель внедрения и бизнес-эффект отдельной фичи. До поручения согласовывать границы, после результата подтверждать именно этот объём; новый запрос не маскировать под исправление дефекта.

Результат сохраняется в предусмотренном рабочем документе или событии системы. Отдельный отчёт каждого участника по каждой работе не требуется. На PoC регулярное взаимодействие с агентами ведёт ИИ-инженер, специалисты обучаются и подключаются через него; на Пилоте работают непосредственно.

Подготовка

PREP-01. Собрать команду внедрения

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Предложить специалистов продукта и выделить им время для решений и проверок.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. В паспорте отражены реальные участники и доступность, включая замещения.

PREP-02. Оформить цели и границы внедрения

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Описать нужное изменение производства ПО и продуктовые ограничения; согласовать границы со Спонсором.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Цель внедрения не заменена бизнес-эффектом отдельной фичи или процентом использования агентов.

PREP-29. Принять решение о переходе к PoC

Владелец продукта — соисполнитель. Ответственный по работе — Спонсор. Работа, результат и предшественники.

1. Подтвердить область выбранных работ и участие команды на встрече по гейту.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Предметные ограничения представлены Спонсору до решения.

PREP-05. Организовать участие и коммуникации

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Обеспечить участие нужных специалистов, замещения и время для ответов.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Ответы не оставлены вне рабочих приоритетов команды.

PREP-30. Подтвердить доступы участников команды

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Лично проверить нужные операции чтения, обсуждения и приёмки под своей учётной записью.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверен действительный доступ, а не только получение приглашения.

PREP-03. Выбрать и назначить задачи PoC в целевом трекере

Владелец продукта — ответственный. Ответственный по работе — Владелец продукта. Работа, результат и предшественники.

1. После проверки инфраструктуры выбрать задачи в целевом трекере с запасом на несколько циклов исправлений.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Есть ссылки на задачи; срочные работы не переданы агентам ради формального охвата.

PREP-04. Согласовать сроки и работы PoC

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Подтвердить реалистичность сроков PoC и участия предметных специалистов.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Пробы и доступность людей совместимы с обязательствами продукта.

PoC

POC-01. Согласовать область и режимы работы Пилота

Владелец продукта — ответственный. Ответственный по работе — Владелец продукта. Работа, результат и предшественники.

1. Выбрать стартовую область Пилота по результатам проб, типам задач и ограничениям.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Непроверенные возможности не обещаны всему продуктовому бэклогу.

POC-02. Подтвердить состав команды и время на Пилот

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Подтвердить время специалистов на прямую работу с агентами и необходимое сопровождение.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Обязанности Пилота обеспечены ресурсом продукта.

POC-15. Определить проверки и приёмку результатов

Владелец продукта — соисполнитель. Ответственный по работе — Тестировщик. Работа, результат и предшественники.

1. Указать, кто принимает продуктовый результат и какие критерии подтверждает.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Техническое одобрение MR и продуктовая приёмка различаются.

POC-21. Согласовать метрики и правила мониторинга

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Связать выбранные метрики с целью продукта и согласовать предварительные ориентиры и баланс.

2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Достижение скорости не рассматривается отдельно от качества, нагрузки и затрат.

POC-22. Рассчитать и согласовать с РП оценочные значения для сравнения

Владелец продукта — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Подтвердить учитываемый состав работ и обоснованность сравнения объёма.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Оценка относится к одинаковым границам результата с агентами и без них.

POC-24. Проверить пробную недельную сводку

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить, что пробная сводка позволяет судить о выбранной цели, а ограничения данных названы.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Успешная отдельная задача не выдана за устойчивый эффект.

POC-03. Согласовать условия и работы Пилота

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Подтвердить приоритет и доступный поток первой недели; уточнить область в паспорте.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Конкретные задачи назначаются в трекере, а не фиксируются заранее в паспорте.

POC-04. Подготовить материалы и провести встречу по гейту

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. На встрече подтвердить продуктовую область, ограничения и участие людей.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Технический успех PoC не подменяет готовность производственного потока.

Пилот

PIL-01. Уточнить критерии и период итоговой оценки

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Подтвердить область, период и актуальные показатели; изменения цели согласовать по полномочиям.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Условия оценки имеют версию и не меняются молча после получения числа.

PIL-02. Проводить ежедневный и недельный разбор внедрения

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Разрешать конфликт приоритетов и обеспечивать задачи с запасом времени на исправления.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Пилот получает рабочий поток, а не только срочные исключения.

PIL-05. Обеспечить освоение обязанностей участниками Пилота

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

Освоение подтверждают действиями специалиста в своей роли и с учётом фактической помощи. Проверка маршрута относится к PIL-10; отдельное учебное задание для отчёта не требуется.

1. Поддерживать достаточный поток подходящих задач и участие специалистов продукта.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Смена исполнителя на агента не исключает продуктовую приёмку.

PIL-06. Закрепить дальнейшее сопровождение и развитие агентов

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Подтвердить людей и ресурс дальнейшей поддержки, учитывая доступную помощь внедрения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. После Пилота не остаётся функций без назначенного исполнителя.

PIL-13. Проверять изменения режима человеческого контроля

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Согласовать ограниченную область изменения контроля и допустимые последствия.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Разрешение не распространено на другие типы задач автоматически.

PIL-15. Согласовывать оценочные значения с РП

Владелец продукта — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Подтвердить существенные изменения состава задачи при уточнении оценок.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Расширение требований отделено от исправления исходного дефекта.

PIL-16. Проверять недельные сводки и полноту данных

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Уточнить продуктовую сторону выбранных аномалий сводки.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Недостаток данных не заменён благоприятной интерпретацией.

PIL-19. Подвести количественные итоги по целям внедрения

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Подтвердить состав выполненного объёма и ограничения продукта в итоговых показателях.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Прогноз и фактически принятый результат различаются.

PIL-21. Описать проверенный способ работы и границы его применения

Владелец продукта — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Зафиксировать типы задач и продуктовые ограничения проверенного способа работы.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Описание не обещает пригодность для ещё не испытанных областей.

PIL-03. Подвести итоги и зафиксировать статус Пилота

Владелец продукта — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Подтвердить фактический продуктовый результат и границы завершённой области.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Принятые и выпущенные результаты не смешаны с незавершёнными.

Связь с процессами и источниками

Порядок совместных действий раскрывают гайды управления внедрением, подготовки команды, технической готовности, работы с агентами и управления эффективностью. Показатели и вычисления задаёт план мониторинга; выполненные улучшения — журнал улучшений.

Гайд роли: РП

Организует выполнение работ внедрения, участие специалистов, решения по этапам и разбор показателей. Не выполняет вместо специалистов все технические проверки и не заменяет решения владельца продукта или Спонсора.

Назначение и распределение работ

РП внедрения организует работы выделенной команды; РП продукта обеспечивает время и действия производственной команды. Оба используют одни паспорт, трекер, встречи и сводки; конкретная работа закрепляется за одним человеком. При совмещении сохраняется результат и решение, отдельная передача самому себе не требуется.

Конкретные участники, принадлежность к команде и выделенное время — в паспорте проекта. У каждой работы один ответственный; соисполнитель подтверждает свой вклад и не наследует все полномочия ответственного. ID и непосредственные зависимости берутся из чек-листов; порядок чтения ниже не создаёт новых зависимостей. Существующий подходящий результат проверяется и используется по ссылке.

Как выполнять работу роли

Ежедневно разбирать зависшие вопросы и требовать содержательной реакции в течение двух рабочих часов. Еженедельно обсуждать показатели и одну-две группы причин. К встрече готовить вопрос, основания и варианты решения; после неё — существующую запись решения и конкретную работу с ответственным и сроком. Не превращать мониторинг в ручное заполнение часов.

Результат сохраняется в предусмотренном рабочем документе или событии системы. Отдельный отчёт каждого участника по каждой работе не требуется. На PoC регулярное взаимодействие с агентами ведёт ИИ-инженер, специалисты обучаются и подключаются через него; на Пилоте работают непосредственно.

Подготовка

PREP-01. Собрать команду внедрения

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Сопоставить нужные роли с реальными специалистами продукта и внедрения; подтвердить выделенное время и замещения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. В паспорте нет общей строки «команда» вместо исполнителей и нет обязательств без доступного времени.

PREP-02. Оформить цели и границы внедрения

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. С владельцем продукта описать нужное изменение, границы и ограничения; подготовить предложение Спонсору.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. На Подготовке зафиксирована содержательная цель без формул и численных обещаний.

PREP-28. Подготовить материалы и провести встречу по гейту

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Собрать подтверждения каждого условия, отделив заявки и обещания от результатов; подготовить повестку решения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. На встрече видны незакрытые условия и конкретные варианты, а не только процент выполненных задач.

PREP-29. Принять решение о переходе к PoC

РП — соисполнитель. Ответственный по работе — Спонсор. Работа, результат и предшественники.

1. Представить Спонсору подтверждённую готовность и ограничения; сохранить решение и условия начала либо повторной проверки.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Дата в календаре сама по себе не разрешает начало PoC.

PREP-05. Организовать участие и коммуникации

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Зафиксировать рабочий чат, ежедневные встречи, еженедельный доклад Спонсору и ответ за два рабочих часа.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Для обращения известны адресат и замещение; зависший вопрос попадает на ежедневную встречу.

PREP-15. Описать общий процесс задач

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. С участниками восстановить существующие статусы, переходы и правила поручения работ.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Описание отражает фактический процесс; подробные правила агентов уточняются на PoC.

PREP-06. Провести вводный семинар для участников PoC

РП — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Обеспечить участие назначенных специалистов во вводном семинаре и доступность материалов.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Сохранены состав участников, демонстрация и предусмотренное учебное упражнение, без требования самостоятельного PoC.

PREP-12. Подготовить рабочие места и доступы людям

РП — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Дать DevOps список людей и операций, действительно нужных их ролям; согласовать получение доступов.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Личный аккаунт инженера не подменяет доступы остальных участников.

PREP-30. Подтвердить доступы участников команды

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Назначить каждому участнику проверочные действия и собрать подтверждения; проблемы направить DevOps.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Факт выдачи аккаунта не считается подтверждением входа и выполнения нужной операции.

PREP-04. Согласовать сроки и работы PoC

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Соотнести сроки паспорта с работами чек-листа и доступностью участников; назначить ближайшие работы в трекере.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Нет отдельного плана PoC и недоступных исполнителей у критичных работ.

PoC

POC-06. Обучить команду и продемонстрировать работу с агентами

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Организовать демонстрацию инженером на задачах продукта, материалы и разбор действий каждой роли.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Специалисты обучены; самостоятельная практическая работа на PoC не сделана обязательным условием.

POC-01. Согласовать область и режимы работы Пилота

РП — соисполнитель. Ответственный по работе — Владелец продукта. Работа, результат и предшественники.

1. Согласовать область Пилота по проверенным возможностям; исключить непроверенные типы работ и зафиксировать ограничения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Успех одной задачи не распространён на весь бэклог и не отменяет HITL.

POC-02. Подтвердить состав команды и время на Пилот

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Уточнить участников Пилота, время на прямую работу с агентами и ресурс сопровождения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Переход от наблюдения к регулярным проверкам учтён в доступности специалистов.

POC-07. Уточнить инфраструктуру и доступы для Пилота

РП — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Обеспечить согласование изменившихся систем и ресурсов, дать актуальный состав пользователей.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Временный контур PoC не принят за готовый контур Пилота.

POC-09. Настроить связность и выдать доступы

РП — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Подтвердить нужные операции людей и интеграций; организовать согласование прав.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Права соответствуют назначенным работам и корпоративным ограничениям.

РОС-12. Организовать личную проверку доступов участников

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Обеспечить личную проверку операций всеми участниками, в том числе в новых инструментах.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Подтверждения относятся к реальным учётным записям Пилота.

РОС-13. Уточнить общий жизненный цикл задач и Git-процесс

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Связать общий ЖЦ задач, Git, способы поручения и правила возврата; согласовать различия РоС и Пилота.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Нет отдельного скрытого процесса агентов и непроверенных контактов с людьми.

РОС-14. Описать точки участия человека

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. В ЖЦ задач закрепить необходимые роли адресатов, материалы и условия продолжения в точках HITL. Конкретных исполнителей и замещения подключить из паспорта по РОС-02, не создавать второй список назначений.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Каждое обращение приходит в чат с событием и материалами; ожидание можно восстановить.

РОС-16. Обеспечить передачу задачи от агента человеку

РП — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Подтвердить порядок смены исполнителя и сохранения работы при передаче человеку.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Команда может забрать задачу без обязательного объяснения; передача не считается успехом агента.

РОС-17. Проверить рабочий маршрут в контуре Пилота

РП — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Устранить организационные задержки сквозного прогона и получить решения участников по их действиям.

2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверены контакты целевого процесса; проба HOTL не добавлена к гейту Пилота.

POC-21. Согласовать метрики и правила мониторинга

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. С владельцем продукта выбрать метрики результата, процесса и баланса, текущую базу и предварительные ориентиры.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. План охватывает HITL и внутренние ожидания; источники и ограничения обозначены.

POC-22. Рассчитать и согласовать с РП оценочные значения для сравнения

РП — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Проверить состав работ и основания оценок объёма и участия людей; подтвердить или уточнить значения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Оценки отличаются от фактов; причина и версия уточнения сохранены.

POC-23. Настроить и проверить ежедневный сбор сведений

РП — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Закрепить суточное расписание или ежедневную просьбу агенту, порядок сбора пропусков.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Человек задаёт период, сведения собирает агент; повторный сбор не удваивает данные.

POC-24. Проверить пробную недельную сводку

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить пробную сводку на выбранных расхождениях и полноте; назначить устранение обязательных ошибок.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Значения PoC не выданы за базу Пилота без проверки сопоставимости.

POC-03. Согласовать условия и работы Пилота

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Актуализировать сроки и условия в паспорте, ориентиры в плане мониторинга, ближайшие работы в трекере.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Первая неделя имеет доступный поток и исполнителей; отдельный план внедрения не создан.

POC-04. Подготовить материалы и провести встречу по гейту

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Свести результаты PoC, готовность контура Пилота и состояние гейта; провести встречу с вариантами решения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверенные возможности, прогноз и непроверенные предположения различаются.

POC-05. Принять решение о переходе к Пилоту

РП — соисполнитель. Ответственный по работе — Спонсор. Работа, результат и предшественники.

1. Сохранить решение Спонсора и назначить действия по его условиям.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Отметки гейта не выданы за отдельное разрешение начать Пилот.

Пилот

PIL-01. Уточнить критерии и период итоговой оценки

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Уточнить окно наблюдения, условия успеха и редакцию плана мониторинга; изменения содержательной цели вынести Спонсору.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Прежние и новые ориентиры и основания пересмотра сохраняются.

PIL-02. Проводить ежедневный и недельный разбор внедрения

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. На ежедневной встрече разбирать зависшие вопросы; еженедельно выбирать приоритетные группы проблем и проверять прошлые изменения.

2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. У решения есть работа, один ответственный и срок; отчёт не заменяет действие.

PIL-05. Обеспечить освоение обязанностей участниками Пилота

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

Освоение подтверждают действиями специалиста в своей роли и с учётом фактической помощи. Проверка маршрута относится к PIL-10; отдельное учебное задание для отчёта не требуется.

1. Обеспечить регулярный поток продуктовых задач, прямое участие специалистов и доступность помощи инженера.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Команда ведёт работу, а не только наблюдает за действиями внедрения.

PIL-06. Закрепить дальнейшее сопровождение и развитие агентов

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Закрепить поддержку платформы, сред, мониторинга и развития за конкретными участниками с ресурсом.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Полная передача инженеру продукта не обязательна; фактические исполнители улучшений видны.

PIL-09. Актуализировать требования к инфраструктуре и инференсу по результатам Пилота

РП — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Согласовать потребность в итоговых ресурсах и стоимости по фактической нагрузке.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. В сравнении использована актуальная конфигурация, изменения не потеряны.

PIL-10. Вести общий процесс задач и Git

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверять соблюдение общего маршрута и актуализировать правила по подтверждённым изменениям.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Задачи, проверки и решения не уходят в недоступные агенту личные договорённости.

PIL-14. Ежедневно собирать сведения и вычислять метрики

РП — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. При ручном способе ежедневно поручать агенту сбор за прошлый день; пропуски догружать тем же навыком.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. В сводке видны покрытие и ошибки, недоступный источник не становится нулём.

PIL-11. Обеспечивать передачу задач от агентов людям

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Организовать сохранение работы и прекращение действий агента при передаче; проверить, что человек может продолжить задачу. Количество, причины и затраты передач рассматриваются в общем цикле эффективности, а не повторно в этой работе.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. После передачи агент остановлен, изменения и результаты доступны человеку; ручное продолжение связано с исходной задачей.

PIL-12. Поддерживать работоспособность обращений HITL

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. По зафиксированным обращениям проверить событие, адресацию по роли, материалы, доступность ответа агенту и переход задачи. Организовать исправление ошибки и повторную проверку цепочки; нехватку времени, замещения или навыка разрешать через PIL-05.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверен полный обмен материалами и продолжение работы; ожидание занятости человека не выдано за техническую ошибку обращения.

PIL-13. Проверять изменения режима человеческого контроля

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Ограничить область пробы контроля, согласовать показатели и возврат; назначить наблюдение после изменения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. HOTL не включается автоматически по сроку; поздние исправления остаются в затратах.

PIL-15. Согласовывать оценочные значения с РП

РП — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Проверять оценки в составе сводки, выделять аномалии и сохранять согласования.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Изменение идеальных часов не названо улучшением производительности само по себе.

PIL-16. Проверять недельные сводки и полноту данных

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить выбранные строки недельной сводки до первичных сведений и согласовать исправления.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Неизвестные исходы, открытые ожидания и слабая выборка остаются видимыми.

PIL-17. Улучшать агентов и способ работы по выявленным ограничениям

РП — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Выделить ресурс на выбранное улучшение организации работы и устранить межролевую зависимость.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. В журнале есть фактический исполнитель и проверка эффекта, а не только новая инструкция.

PIL-19. Подвести количественные итоги по целям внедрения

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Свести показатели по актуальным целям и сопоставимым границам; сохранить исходные наборы и версии.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проценты разных периодов не усреднены вместо расчёта по наблюдениям.

PIL-20. Проверить качество, нагрузку людей и стоимость

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить показатели баланса и полную принятую границу затрат вместе с результатом.

2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Ускорение не оплачено скрытым ростом ручной помощи или расходов.

PIL-21. Описать проверенный способ работы и границы его применения

РП — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Подтвердить фактическую организацию работы, назначения и границы применимости.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Описан реально проверенный режим, а не намерение повысить автономность.

PIL-22. Устранить блокирующие отклонения и подтвердить результат

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Выбрать обязательные для согласованных условий отклонения, назначить исправления и повторную проверку.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Закрытие задач без подтверждения результата не закрывает условие Пилота.

PIL-03. Подвести итоги и зафиксировать статус Пилота

РП — ответственный. Ответственный по работе — РП. Работа, результат и предшественники.

1. Собрать итоговый срез и подтверждения обязательных условий; при выполнении зафиксировать успех и сообщить участникам.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Отдельного одобрения достигнутых условий Спонсором не требуется.

PIL-04. Определить дальнейшие действия при невыполнении условий Пилота

РП — соисполнитель. Ответственный по работе — Спонсор. Работа, результат и предшественники.

1. При отклонениях представить Спонсору варианты продолжения, изменения условий с продолжением или остановки.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. После решения обновлены сроки, работы и условия проверки; успех не объявлен заранее.

Связь с процессами и источниками

Порядок совместных действий раскрывают гайды управления внедрением, подготовки команды, технической готовности, работы с агентами и управления эффективностью. Показатели и вычисления задаёт план мониторинга; выполненные улучшения — журнал улучшений.

Гайд ИИ-инженера

ИИ-инженер обеспечивает технические условия совместной работы команды с агентами, адаптирует агентов к продукту, поддерживает их развитие и расчёт показателей внедрения. На Подготовке он проверяет рабочий контур; на PoC ведёт пробы и доработки с участием специалистов продукта; на Пилоте помогает команде, которая сама ведёт поток задач.

Регулярные разборы, выбор изменений и проверка их эффекта организуются по гайду процесса управления эффективностью. Здесь описаны действия роли в этом процессе.

Гайд раскрывает работы трёх чек-листов, в которых ИИ-инженер назначен ответственным или соисполнителем. ID работ сохранены. У каждой работы остаётся один ответственный: участие инженера не переносит на него полномочия РП, владельца продукта, DevOps или проверяющих. На роль может быть назначено несколько человек с распределёнными работами. Один человек может совмещать роли, например РП и ИИ-инженера. Результат работы фиксируется в предусмотренном документе или рабочей системе и сохраняется независимо от распределения ролей. Согласование относится к полномочию роли: при совмещении сохраняется запись проверки или решения, отдельная передача самому себе не требуется.

В ссылке у работы находятся её результат, предшественники и связь с гейтом. Здесь подробно описаны действия инженера и способ проверки своего вклада. Порядок подразделов помогает читать этап, но не вводит новые зависимости: независимые части выполняются параллельно. Уже существующий подходящий результат проверяется и используется по ссылке; повторять работу только ради нового документа не нужно.

Кто выполняет роль. ИИ-инженер может быть специалистом команды продукта, команды внедрения или предприятия. Команда внедрения может быть внешним консультантом, командой поставщика либо внутренней группой. Принадлежность не меняет содержание роли. Для порученной работы нужны знания платформы и способа адаптации агентов, доступ к инструкциям, навыкам и диагностике; знания продукта получаются с участием его специалистов.

Назначение работ. В общей методологии указывается роль. В разделе команды паспорта — конкретный участник, его роль, команда и выделенное время. Закреплённые работы указываются в плане; обязанности и полномочия следуют из роли. Один человек может совмещать ИИ-инженера и РП; несколько ИИ-инженеров могут разделить работы. Агент внедрения предлагает назначения по подсказкам, а подтверждённое распределение фиксируется в плане. Инженер команды внедрения не получает автоматически полномочий выделять ресурсы предприятия или менять системы продукта. ИИ-инженеру внедрения может быть поручена начальная настройка и обучение ИИ-инженера продукта; конкретные работы и фактическая передача фиксируются в плане и журнале.

> *Если участвует команда внедрения*
>
> **Специалист внедрения.** ИИ-инженер внедрения первоначально настраивает и адаптирует агентов, разбирает сложные случаи, настраивает мониторинг и обучает ИИ-инженера продукта на задачах проекта.
>
> **Специалист продукта / предприятия.** ИИ-инженер продукта осваивает диагностику, улучшение навыков и сопровождение агентов и принимает согласованные работы по мере готовности. Часть работ может оставаться у специалиста внедрения.

На PoC ИИ-инженер ведёт регулярное взаимодействие с агентами. При подключении специалистов продукта на последних задачах он предварительно проверяет материалы и обратную связь, сопровождая обмен в обе стороны. Команда проходит обучение и наблюдает демонстрацию инженера; самостоятельная работа и обязательная практическая проба на PoC не требуются. На Пилоте специалисты продукта взаимодействуют с агентами непосредственно, инженер помогает при затруднениях и развивает агентов.

Подготовка

Подготовить доступный технический путь и условия начала PoC. До гейта проверяется инфраструктура, а не предметная способность агента решать задачи продукта. Метрики и полноценный мониторинг готовятся на PoC.

Чек-лист и работы этапа.

PREP-02. Оформить цели и границы внедрения

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Из границ продукта и предполагаемых видов работ выделить технические условия: доступные исходники, способы сборки, проверок и развёртывания, внешние зависимости, ограничения доступа и ИБ. Разделить подтверждённое, требующее проверки и пока недоступное.
2. В паспорте проекта описать ограничения первоначальной области: например, изменение сервиса доступно агенту, а проверка интеграции требует управляемого мока. Указать выполнимый начальный объём и возможности, которые ещё предстоит проверить.
3. В паспорте проекта сохранить технические ограничения и ссылки на источники. Бизнес-цели формулирует владелец продукта, согласование организует РП.
4. В паспорте сохранить технические вопросы, которые предстоит проверить на PoC. На Подготовке цель формулируется по сути; метрики, формулы и численные ориентиры не выбираются. Их выбор и подготовка плана мониторинга относятся к POC-21, исходные оценки — к POC-22.

Результат работы инженера. В паспорте отражены технические ограничения выбранной области; известно, какие условия нужно обеспечить до PoC. Цель и границы описаны содержательно, вопросы для проверки на PoC сохранены.

PREP-05. Организовать участие и коммуникации

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Для назначений и взаимодействия в паспорте указать, участие каких ролей и какое время понадобятся на PoC и Пилоте: требования, вопросы о коде, MR, тесты, продуктовые решения. Конкретных людей и замещения закрепляет РП. Подробные события и условия продолжения оформляются в ЖЦ задач, а не дублируются в назначениях.
2. Для обучения и демонстраций определить, какие действия на PoC воспроизводит инженер и когда могут подключаться коллеги. На Пилоте ориентироваться на непосредственную работу профильных специалистов с доступной помощью. В пробах указывать фактического участника; воспроизведение инженером не доказывает освоения работы специалистом продукта.

3. Обеспечить агентам доступ ко всем предусмотренным видам коммуникаций PoC и Пилота: постановкам, замечаниям, ответам и значимым решениям встреч. Необходимые подключения и права записать в карточке инфраструктуры, каналы и место сохранения сообщений — в описании взаимодействий и HITL. Это техническое обеспечение согласованного участия; подробно маршрут уточняется и проверяется в PREP-15/16 и POC-13/14/17. При отличии систем PoC и Пилота сохранить оба варианта и необходимые изменения.
4. Указать в паспорте проекта необходимую помощь инженера и способ обращения за ней. РП организует ежедневные встречи, ответ в течение двух часов рабочего времени и решение вопросов загрузки участников.

Результат работы инженера. В паспорте отражены необходимые участие, помощь и условия обучения; средства предусмотренных коммуникаций доступны, подключения описаны в инфраструктуре. Проектирование событий и переходов не подменяется назначением людей.

PREP-07. Оформить и согласовать карточку инфраструктуры

ИИ-инженер — ответственный. Работа и результат. Соисполнитель — DevOps.

Подсказка для назначения. Ответственным может быть ИИ-инженер команды внедрения или продукта, который сводит общую конфигурацию. Нужны знания платформы, доступ к сведениям об инфраструктуре и участие уполномоченных специалистов продукта. При нескольких DevOps в проекте их вклады — требования установки, ресурсы, сети и доступы — указываются в плане отдельно. Принадлежность к команде сама по себе не назначает ответственного.

1. В разделе «Команда проекта» паспорта зафиксировать назначенного ИИ-инженера и участников DevOps с их командами и выделенным временем. Конкретные работы, ответственного и вклад соисполнителей назначить в трекере по чек-листу этапа. Роль ИИ-инженера остаётся общей.
2. В карточке инфраструктуры свести инференс, ИИ-платформу, трекер, Git, CI/CD, анализ кода и артефакты по применимости, чат, документацию и тестовые среды. Указать размещение, адреса, требования к ресурсам и ответственные за системы стороны.
3. В той же карточке инфраструктуры описать необходимые соединения, аутентификацию и операции; разделить права людей, систем и платформы. DevOps продукта/предприятия уточняет доступные ресурсы и ограничения своей инфраструктуры; DevOps, назначенный на установку, — требования поставки и доступа к месту установки.
4. Для on-prem указать ресурсы и доступ, которые должны быть предоставлены до установки, и результат их проверки. Если предоставление и установку выполняют разные участники, в адаптированном плане оформить отдельные работы со ссылкой на PREP-08 и зависимостью установки от готовности ресурсов. Для облака или существующего экземпляра указать применимые действия подключения, не создавая ненужных работ установки.
5. Отметить временные компоненты PoC и отличия целевого контура Пилота. В карточке инфраструктуры сохранить предусмотренные проектом технические согласования и согласование ИБ. Фактическая настройка и проверка подключений фиксируются последующими работами в той же карточке.

Результат работы инженера. Согласованная карточка инфраструктуры описывает конфигурацию и доступы; в проектном плане есть подтверждённые назначения, разделение предоставления инфраструктуры и установки по применимости и необходимые зависимости.

PREP-08. Развернуть или подключить платформу

ИИ-инженер — соисполнитель. Ответственный — DevOps. Работа и результат.

Подсказка для назначения. На установку назначается DevOps с компетенцией развёртывания платформы — из команды внедрения или продукта. На выделение ресурсов и выдачу доступа для установки нужен участник с соответствующими полномочиями в инфраструктуре продукта/предприятия. Если эти части разделены между людьми, у каждой в адаптированном плане свой ответственный. ИИ-инженер проверяет работоспособность функций агента, а не подменяет ответственность за предоставление ресурсов или установку.

1. По карточке инфраструктуры и проектному плану определить применимый вариант: новая on-prem установка, облачный проект или подключение действующего экземпляра. Сверить конкретных исполнителей установки и предоставления ресурсов, не предполагая, что все действия выполняет один DevOps.
2. Для on-prem в задаче установки использовать подтверждение предоставленных ресурсов и доступа. Если их нет, сохранить незавершённое условие и связанную работу; назначение DevOps команды внедрения не означает, что инфраструктура уже предоставлена. Для облака или существующей платформы использовать запись о созданном или подключённом проекте.
3. В карточке инфраструктуры сверить записанные DevOps адрес, версию, проект и способ входа. Из рабочего контура проверить создание и запуск сессии, инструменты, сохранение журналов и получение результата под предназначенными для PoC полномочиями.
4. В задаче установки или подключения сохранить выявленные ошибки: операция, контур, время, диагностика. После исправления назначенным DevOps повторить затронутые проверки и сохранить результаты в карточке инфраструктуры.

Результат работы инженера. В карточке инфраструктуры подтверждены функции платформы для PoC; проверки связаны с конкретным проектом и выполненной установкой или подключением, а незавершённые инфраструктурные условия отражены в проектных работах.

PREP-09. Подключить и проверить инференс

ИИ-инженер — ответственный. Работа и результат.

1. Уточнить поддерживаемые платформой подключения и назначения моделей. Подключить основную текстовую модель и модель зрения; это может быть одна модель. Для других назначений указать, нужны ли они выбранным работам.
2. Из фактического контура платформы проверить вызовы, получение ответа и обработку ошибки подключения. Затем проверить применение модели через рабочую сессию агента: успешный отдельный HTTP-вызов не подтверждает всю цепочку.
3. Зафиксировать в карточке инфраструктуры модели и версии, адреса, доступные лимиты, условия использования и результат проверки. Секреты хранить средствами платформы, в карточке оставлять только сведения о способе подключения.
4. При ограничении мощности или доступных функций указать влияние на параллелизм и виды работ. С DevOps устранить техническую причину либо в паспорте проекта зафиксировать ограничение и согласованное изменение области.

Результат работы инженера. Вызовы требуемых моделей работают из платформы; сохранены конфигурация, лимиты и подтверждения.

PREP-10. Организовать хранение материалов внедрения

ИИ-инженер — ответственный. Работа и результат.

1. Подготовить репозиторий или согласованное место материалов внедрения. Разместить ссылки на паспорт, карточку команды, карточку инфраструктуры и чек-лист готовности к Пилоту; подключать существующие записи ссылками, а не копировать факты.
2. Определить, где хранятся проектные инструкции, навыки и память агентов. Проверить версионирование материалов, доступ инженера и возможность чтения агентом там, где это потребуется.
3. Создать понятную начальную навигацию без требования заранее заполнить индекс всего контекста продукта. Достаточно доступного места для сведений, которые будут появляться при работе.

Результат работы инженера. Инженер может читать и сохранять материалы, история изменений доступна; место будущих инструкций и памяти определено.

PREP-11. Настроить связи и доступы системам и ИИ-платформе

ИИ-инженер — соисполнитель. Ответственный — DevOps. Работа и результат.

1. В карточке инфраструктуры перечислить необходимые операции компонентов платформы для настройки DevOps. Уточнить различия между правами человека в браузере и служебной учётной записью агента.
2. Проверить подключённые инструменты из контура платформы: сеть, аутентификацию, чтение и разрешённые операции изменения. Для каждой проверки сохранить систему, контур, операцию и результат без секретов.
3. При отказе различить отсутствие связи, ошибку аутентификации, недостаток прав и ошибку самого инструмента. В задаче настройки доступа сохранить диагностику; после исправления повторить операцию и записать результат проверки.

Результат работы инженера. Подключения компонентов платформы проверены фактическими операциями; карточка инфраструктуры содержит результаты, а не только выданные права.

PREP-12. Подготовить рабочие места и доступы людям

ИИ-инженер — соисполнитель. Ответственный — DevOps. Работа и результат.

1. Описать DevOps своё рабочее окружение: доступ к проекту платформы, локальный агент при выбранной схеме, Git, трекер, журналы и средства диагностики.
2. Проверить, что из этого окружения можно выполнить необходимую инженерную работу: открыть журналы, изменить инструкцию в выбранном месте и запустить проверку. Уточнить недостающие права, не заменяя их общей привилегированной учётной записью.
3. В карточке инфраструктуры записать результат своей проверки; обнаруженные ошибки оформить задачами со ссылками на диагностику. Личные проверки остальных участников организует РП; инженер не подтверждает их доступ только по факту создания аккаунтов.

Результат работы инженера. В карточке инфраструктуры описаны требования к рабочему окружению, сохранены проверки операций инженера и ссылки на задачи устранения проблем.

PREP-16. Описать общий Git-процесс

ИИ-инженер — соисполнитель. Ответственный — Разработчик. Работа и результат.

1. Разобрать с разработчиком путь изменения: от какой ревизии создаётся ветка, как появляется MR, что запускает CI, где выполняется ревью и кто делает merge.

2. Для каждой операции проверить, может ли её выполнить агент имеющимися инструментами и правами. Отдельно отметить действия человека и ограничения защищённых веток, CI и развёртывания.
3. В описании Git-процесса зафиксировать проверенные операции и ограничения агента; необходимые изменения оформить замечаниями к этому описанию. На Подготовке фиксируется действующий процесс; подробные правила взаимодействия с агентами будут уточняться на PoC.

Результат работы инженера. В описании Git-процесса отражены ограничения операций агентов, не создан параллельный процесс вне рабочего Git.

PREP-13. Проверить технический путь сборки и развёртывания

ИИ-инженер — соисполнитель. Ответственный — DevOps. Работа и результат.

1. Получить от разработчика и DevOps существующую ревизию, команды и ожидаемые результаты технического прогона. Для этой проверки не требуется заранее придумывать задачу онбординга.
2. Из рабочего контура платформы проверить доступ к средствам сборки и развёртывания. Проследить связь исходной ревизии, сборки, неизменяемого ID артефакта, развёрнутой версии и контрольной проверки.
3. Убедиться, что доступны журналы неуспешной сборки или проверки. В задаче технического прогона сохранить ошибки и диагностические материалы; после исправления повторить затронутые действия и записать результат.

Результат работы инженера. Технический путь доступен из контура платформы; успешная сборка с ноутбука человека не используется как единственное подтверждение.

PREP-14. Подготовить тестовые среды продукта

ИИ-инженер — соисполнитель. Ответственный — DevOps. Работа и результат.

1. С DevOps и тестировщиком проверить, как агент выбирает среду и узнаёт фактическую ревизию, подготавливает данные, запускает e2e и получает результаты и диагностику.
2. По зависимостям выбранной области проверить программное управление: ответы и ошибки моков, эмуляторы оборудования, виртуальное время, сброс состояния и данных. Для требуемых действий должны быть доступны команды, API или поддерживаемые инструменты агента.
3. Проверить доступ агента к среде приёмки для разбора замечаний. Зафиксировать ручные действия и отличия от продуктивной среды; необходимые доработки оформить задачами с ответственными, результаты проверок внести в карточку инфраструктуры.

Результат работы инженера. Агенту доступны средства управления тестовой средой и диагностики; тестирование не зависит от отсутствующего у него персонального компьютера или физического оборудования.

PREP-27. Подтвердить готовность систем и связности

ИИ-инженер — ответственный. Работа и результат.

1. Пройти по необходимым системам и связям из карточки инфраструктуры, используя записи выполненных проверок. Для каждой сверить учётную запись, рабочий контур и разрешённые операции.
2. Проверить связанный технический путь: инференс, инструменты платформы, исходники, сборка, артефакт, развёртывание, тесты и получение результата. Уже подходящие проверки

использовать по ссылкам; изменившиеся подключения проверить заново.

3. С DevOps закрыть выявленные технические ошибки, с тестировщиком подтвердить проверки среды. В карточке инфраструктуры записать итог готовности либо конкретные незавершённые работы; открытая заявка не является успешной проверкой.

Результат работы инженера. Готовность систем и связности подтверждена в карточке инфраструктуры. Личные доступы участников отдельно проверяются под организацией РП.

PREP-30. Подтвердить доступы участников команды

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Лично проверить нужные операции в трекере, Git, чате, платформе и средствах диагностики под назначенными учётными записями.
2. Сохранить подтверждения в карточке инфраструктуры; при ошибке назвать операцию, контур и сообщение, не помещая секреты в запись.
3. После исправления DevOps повторить свою проверку. Доступ инженера не подтверждает доступы остальных участников.

Результат работы инженера. Собственные разрешённые операции выполнены и подтверждены; незакрытые проблемы видны РП.

PREP-03. Выбрать и назначить задачи PoC в целевом трекере

ИИ-инженер — соисполнитель. Ответственный — Владелец продукта. Работа и результат.

1. После готовности инфраструктуры посмотреть предложенные владельцем продукта задачи в целевом трекере. Проверить, что их исходники, зависимости и проверки соответствуют подготовленному контуру.
2. В целевом трекере отметить назначение выбранных задач: для второй фазы после онбординга — лёгкие задачи, для третьей — целевые задачи, характерные для будущего потока Пилота. Конкретные пробы онбординга заранее не оформляются как обязательные продуктовые задачи.
3. Если задача требует дополнительной среды или доступа, указать требование до её включения. В материалах проверки готовности к PoC сохранить результат соответствия задач подготовленным условиям; выбор приоритетов остаётся за владельцем продукта.

Результат работы инженера. Выбранные задачи выполнимы в подготовленных условиях и представлены ссылками или фильтром целевого трекера в результатах выбора задач, без отдельного дублирования постановок.

PREP-17. Проверить чтение выбранных задач агентами

ИИ-инженер — ответственный. Работа и результат.

1. Из рабочего контура агента открыть каждую выбранную задачу либо проверить согласованный список через фильтр. Проверить не только заголовок, но и описание, комментарии, необходимые вложения и связанные материалы.
2. Попросить агента кратко изложить прочитанное и сослаться на источники. Вместе с аналитиком проверить, не потерялись ли существенные требования из недоступного формата или вложения.
3. Исправить техническую проблему чтения либо предоставить материал доступным способом. Сохранить ссылки на задачи и результаты в материалах проверки чтения; углублённое уточнение требований выполняется уже на PoC.

Результат работы инженера. Чтение выбранных задач подтверждено агентом и сверено с аналитиком; доступ к списку задач не подменяет доступ к их содержанию.

PREP-04. Согласовать сроки и работы PoC

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. По чек-листу готовности к Пилоту определить технические работы и ожидаемые подтверждения. В назначенных работах отразить порядок онбординга: проба, разбор ошибки, доработка и повторная проверка; указать место сохранения навыков, памяти и результатов.
2. Связать выбранные продуктовые задачи в трекере с проверяемыми возможностями. Конкретные пробы онбординга и подробные постановки уточняются по ходу PoC.
3. Сопоставить объём работ и участие специалистов со сроками в паспорте, отметить технические зависимости и направления проверки с тестировщиком. У конкретной работы один ответственный в трекере. Полную готовность мониторинга обеспечить к Пилоту, а не до онбординга.

Результат работы инженера. Работы PoC назначены и связаны с проверяемыми результатами чек-листа; сроки отражены в паспорте, задачи и пробы — в трекере. Отдельный план не создаётся.

PREP-06. Провести вводный семинар для участников PoC

ИИ-инженер — ответственный. Работа и результат.

1. Подготовить короткую демонстрацию: как поручить работу агенту, где читать обращение, как ответить, передать замечание и найти результат. Использовать доступные инструменты PoC и действующие договорённости с РП.
2. Провести упражнение: агент задаёт вопрос, участник получает его в чате, отвечает, а агент использует ответ. Аналитик, разработчик и тестировщик пробуют действия в своей зоне; объяснение без действий не заменяет упражнение.
3. Разобрать недошедшие уведомления, недоступные материалы и неоднозначные ответы. В материалах вводного семинара сохранить презентацию, состав участников, результаты упражнения и выявленные вопросы; связать их с протоколом встречи.

Результат работы инженера. Участники выполнили свои действия в упражнении, результаты и вопросы сохранены; предметные возможности агента ещё не выдаются за подтверждённые на PoC.

PREP-28. Подготовить материалы и провести встречу по гейту

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Для РП собрать ссылки на технические результаты: карточку инфраструктуры, проверку моделей, подключения, прогон ревизии, среды и чтение задач. Выделить ещё не закрытые условия.
2. Проверить, что подтверждения относятся к выбранному контуру PoC и всему выбранному перечню задач, а не к чужому проекту или временному доступу с компьютера инженера.
3. На встрече представить техническую готовность и ограничения. При отрицательном результате назвать конкретную незавершённую работу, её последствие и необходимую проверку после исправления.

Результат работы инженера. В материалах гейта сохранены ссылки на технические подтверждения; незавершённые условия не скрыты формулировкой «в целом работает».

PREP-29. Принять решение о переходе к PoC

ИИ-инженер — соисполнитель. Ответственный — Спонсор. Работа и результат.

1. На встрече отвечать на технические вопросы Спонсора: что уже проверено, что предстоит адаптировать на PoC и какие ограничения сохраняются.
2. При предложении изменить область в материалах гейта перечислить затронутые подключения и проверки, а согласованные изменения внести в паспорте проекта и карточке инфраструктуры. Не подтверждать новые возможности без проверки.
3. После решения сверить с РП разрешённую область и технические действия: начинать PoC либо продолжать Подготовку. Решение принимает Спонсор, запись решения ведёт РП.

Результат работы инженера. Технические ограничения учтены в решении; инженер действует в согласованной области.

PoC

Через пробы и ошибки научить агентов работать с продуктом и подтвердить выбранную область Пилота. Временные Git, трекер и среды допустимы, но взаимодействия людей и агентов должны быть репрезентативны. При переходе на целевые системы повторно проверяются затронутые условия.

Чек-лист и работы этапа.

POC-18. Адаптировать агентов на задачах PoC

ИИ-инженер — ответственный. Работа и результат.

1. Начать с доступного проекта и минимального рабочего пути. Через агента проверить чтение материалов и кода, получение задачи, изменение, сборку, развёртывание и получение доказательств. Первую пробу подобрать по ситуации; не требовать полного контекста продукта заранее.
2. По вопросам агента обратиться к существующим материалам и профильным специалистам. Проверить сведения на коде или выполнении; сохранить устойчивые правила в проектных инструкциях, например AGENTS.md, повторяемые процедуры — в навыках, найденные факты и ссылки — в памяти агента. Не копировать всё содержание чатов в постоянные инструкции.
3. На каждом затруднении собрать задачу, ревизию, версию навыка, действие инструмента, ожидаемый и фактический результат. Отличить неполное требование, недоступный источник, ошибку навыка, дефект инструмента и недостаток среды. Не исправлять промпт вместо отсутствующего доступа или управляемого мока.
4. Сформулировать точечное изменение. Например, если агент проверил не ту версию, дополнить процедуру проверкой ID артефакта и развёрнутой ревизии; если не понял замечание — уточнить обработку замечания и способ обращения за разъяснением.
5. Повторить проблемный сценарий и зависимые проверки. Зафиксировать, помогло ли изменение, и оставить, доработать либо отменить его. При ручной помощи сохранить, что именно сделал человек; этот результат не считать автономным выполнением агента.
6. Постепенно расширять пробы до выбранных задач и контактов с людьми. Самообучение должно оставлять доступный результат: обновлённую память или инструкцию с источником; инженер проверяет полезность изменений по выполнению, а не по объёму написанного текста.

Результат работы инженера. Есть журнал проб, версии изменённых материалов, результаты проверок и подтверждённые возможности с ограничениями. Контекст появился по фактическим вопросам; отдельный индекс контекста не требуется.

POC-19. Свести результаты проверок PoC

ИИ-инженер — ответственный. Работа и результат.

1. По решённым задачам собрать ссылки на постановки, ревизии, MR, сборки, тесты, замечания и версии инструкций. Для каждого проверенного вида работ указать условия, результат и участие человека.
2. Получить заключения профильных специалистов: аналитик проверяет требования, разработчик — код, тестировщик — качество тестов и доказательств. Технически завершённый запуск без содержательной проверки не выдавать за успешную возможность.
3. В итогах PoC отделить проверенное, непроверенное и неподходящее текущей конфигурации. Не создавать вторые карточки задач; использовать ссылки на результаты в рабочих системах.

Результат работы инженера. Сводка PoC позволяет выбрать область Пилота по доказательствам и видеть, где успех зависел от помощи инженера.

POC-01. Согласовать область и режимы работы Пилота

ИИ-инженер — соисполнитель. Ответственный — Владелец продукта. Работа и результат.

1. По сводке PoC предложить владельцу продукта, какие типы задач и участки процесса подтверждены и какие ограничения остаются. Для каждого предложения дать ссылки на проверенные задачи.
2. С разработчиком и тестировщиком уточнить необходимый контроль и доказательства для стартового потока. Разделять расширение типов задач и изменение контроля; переход к HOTL на этой стадии не требуется.
3. В паспорте проекта описать подтверждённые возможности, технические границы и ограничения стартовой области. Если новый тип задачи или зависимость не проверены, назвать недостающую проверку вместо общего обещания доработать по ходу.

Результат работы инженера. В паспорте проекта зафиксированы подтверждённая область и ограничения; старт Пилота не основан на экстраполяции одной удачной задачи.

POC-07. Уточнить инфраструктуру и доступы для Пилота

ИИ-инженер — соисполнитель. Ответственный — DevOps. Работа и результат.

1. Сравнить временный контур PoC с предлагаемым контуром Пилота. Перечислить изменения адресов, репозиториев, трекера, CI, сред, учётных записей, моделей и лимитов.
2. В карточке инфраструктуры уточнить требования платформы и агентов: операции интеграций, нужные ресурсы, плановый параллелизм и диагностические данные. Если контур сохраняется, проверить его соответствие выбранному потоку.
3. Проверить свою часть обновлённой карточки инфраструктуры. Отдельно отметить, какие результаты проверок PoC остаются применимыми, а какие нужно повторить после смены подключения.

Результат работы инженера. В карточке инфраструктуры учтены изменения для Пилота и необходимые агентам операции; нет предположения, что временный доступ подтверждает целевой.

POC-08. Подготовить системы и ресурсы Пилота

ИИ-инженер — соисполнитель. Ответственный — DevOps. Работа и результат.

1. После подготовки DevOps систем Пилота проверить рабочий проект платформы, инструменты и инференс. Сверить модели, лимиты и окружение с обновлённой карточкой инфраструктуры.
2. Выполнить проверочный запуск сессии и модельных вызовов. При смене экземпляра перенести нужные настройки проекта, сохранив исходные версии и различия.
3. В задачах настройки систем Пилота сохранить ошибки и ссылки на диагностику, повторить проверки после исправления; результаты отразить в карточке инфраструктуры. Отметить техническую проверку отдельно от последующего сквозного выполнения задачи.

Результат работы инженера. Функции платформы и модели доступны в подготовленных системах Пилота; результат сохранён в карточке инфраструктуры.

POC-09. Настроить связность и выдать доступы

ИИ-инженер — соисполнитель. Ответственный — DevOps. Работа и результат.

1. Уточнить DevOps операции для каждой изменённой интеграции: чтение и изменение задачи, ветка и MR, CI, артефакты, среды, чат, документация. Проверить, какой компонент использует каждую учётную запись.
2. Обновить настройки инструментов и ссылки, которые зависят от временных адресов PoC. Секреты перенести разрешённым способом, не помещая их в инструкции и журнал.
3. Проверить подключение после настройки. Учесть отличие прав временного контура: они не должны автоматически считаться допустимыми в целевых системах.

Результат работы инженера. Операции и настройки интеграций соответствуют контуру Пилота; необходимые технические проверки можно выполнить под его учётными записями.

POC-11. Проверить техническую готовность из контура Пилота

ИИ-инженер — ответственный. Работа и результат.

1. Из контура ИИ-платформы пройти по разрешённым операциям Пилота: прочитать и обновить задачу, получить репозиторий, создать изменение и MR, запустить CI, развернуть артефакт по ID, выполнить тесты и получить результат.
2. Отправить обращение в рабочий чат и проверить получение ответа агентом. Для ошибки проверить доступность её диагностики, а не только успешный сценарий.
3. В карточке инфраструктуры сохранить систему, учётную запись, операцию, время и ссылку на доказательство. Исправления DevOps перепроверить; результаты временного контура не переносить вместо новых проверок.

Результат работы инженера. Техническая цепочка Пилота проверена из её фактического контура; обязательных неработающих подключений не осталось.

POC-12. Организовать личную проверку доступов участников

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. По просьбе РП выполнить собственные действия в системах Пилота: просмотр задачи и MR, чтение тестового отчёта, ответ в чате, доступ к журналам и изменение инструкций по своим полномочиям.

2. В карточке инфраструктуры сохранить результаты фактического входа и действий; ошибки оформить задачами настройки доступов. После исправлений повторить соответствующую операцию.
3. Не подтверждать за остальных участников их личные доступы. РП собирает отметки каждого и сохраняет их в карточке инфраструктуры.

Результат работы инженера. В карточке инфраструктуры сохранены проверки собственных доступов инженера и задачи устранения выявленных проблем.

POC-14. Описать точки участия человека

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. По пробам PoC перечислить обращения, которые агент формирует по навыкам: вопрос о требовании, неоднозначное замечание, недостаток данных, ошибка или отсутствие полномочия.
2. Для каждого обращения определить событие, материалы и вопрос, которые попадут в чат. С РП и специалистами связать его с ролью отвечающего и условием продолжения; отдельные правила Git и трекера использовать из общего процесса.
3. Сопоставить контакты людей и агентов на PoC и Пилоте. При временных инструментах сохранить смысл HITL, ревью, приёмки и передачи; отсутствовавшие взаимодействия включить в сквозную проверку.

Результат работы инженера. В маршруте задачи описаны обращения по навыкам и их связь с процессом; запрос содержит наступившее событие и достаточные материалы для ответа.

POC-16. Обеспечить передачу задачи от агента человеку

ИИ-инженер — ответственный. Работа и результат.

1. Настроить прекращение выполнения агентом по решению команды. Проверить, как отменяются текущие действия и предотвращается продолжение изменений после смены исполнителя.
2. Сохранить рабочую ревизию, ветку или MR, незавершённые изменения и диагностику, чтобы человек мог продолжить. С РП уточнить изменение исполнителя и статуса в общем процессе.
3. Обеспечить событие передачи в рабочих системах для будущей метрики. Причина может не указываться: команда вправе забрать задачу без обоснования; на PoC не устанавливаются обязательный перечень оснований и предел попыток.

Результат работы инженера. Разработчик может получить задачу и изменения, агент прекращает выполнение, событие передачи доступно мониторингу.

POC-20. Подготовить и проверить рабочую версию агентов

ИИ-инженер — ответственный. Работа и результат.

1. Зафиксировать исходную рабочую версию инструкций, навыков, памяти и инструментов для старта Пилота. Убрать из неё устаревшие адреса временных систем и проверить актуальные полномочия.
2. Повторить необходимые сценарии выбранной области после исправлений. Для замечаний к коду и тестам получить подтверждение разработчика и тестировщика; ссылку на проверку связать с версией.
3. Сохранить возможность вернуться к предыдущей рабочей версии. Зафиксировать оставшиеся ограничения и исключить из стартовой области сценарии с незакрытым обязательным

замечанием.

Результат работы инженера. Стартовая версия восстанавливается по сохранённым материалам, имеет подтверждения проверок и перечисленные ограничения.

PoC-17. Проверить рабочий маршрут в контуре Пилота

ИИ-инженер — ответственный. Работа и результат.

1. В контуре Пилота выполнить задачу по согласованному маршруту: запуск, выполнение, MR, проверки, запрос HITL, исправление замечания и приёмка. Привлечь специалистов для их реальных действий.
2. Проверить передачу человеку: агент остановился, изменения сохранены, человек продолжает, событие доступно для учёта. Если это не возникло естественно, явно обозначить учебную пробу.
3. Допроверить контакты людей и агентов, которые отсутствовали или существенно отличались на PoC. Сохранить ссылки на задачу, версии правил и результаты. Пробу HOTL в этот прогон не включать: она рассматривается на Пилоте по метрикам.

Результат работы инженера. Рабочий маршрут Пилота подтверждён в целевых условиях, технические и организационные ошибки не скрыты ручной подменой действий агентом или инженером.

PoC-06. Обучить команду и продемонстрировать работу с агентами

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Подготовить материалы нового процесса: как поручать и уточнять задачи, получать обращения, проверять результаты и возвращать замечания. Показать действия аналитика, разработчика и тестировщика, предусмотренные точки HITL и отличия процесса Пилота от PoC.
2. На примере задач продукта продемонстрировать полный обмен: проверку понимания задачи агентом, рассмотрение MR и тестов, формулирование замечаний и повторную проверку исправлений. Объяснить, как действовать при затруднениях и когда обращаться за помощью к ИИ-инженеру.
3. Разобрать вопросы и сохранить материалы, состав участников и ссылку на демонстрацию либо её протокол в подтверждениях обучения команды. Демонстрация в ходе обычной работы засчитывается; отдельное занятие не обязательно. Самостоятельные действия участников и обязательная практическая проба не требуются. При подключении специалистов сопровождать обмен и проверять материалы в обе стороны.

Результат работы инженера. Команде представлены новый процесс и действия в своих ролях, продемонстрирован рабочий цикл; РП может подтвердить проведённое обучение материалами, составом участников и ссылкой на демонстрацию либо её протокол.

PoC-21. Согласовать метрики и правила мониторинга

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Для предложенных РП метрик проверить, какие первичные события доступны в описанном жизненном цикле задач, Git, чате, журналах агентов и тестовых сред. Указать способ связи события с задачей и ограничения источника.
2. Различить регистрируемые факты и величины, которые рассчитываются оценочно: объём работ в идеальных часах и время людей. Для оценок предусмотреть основания, версии и согласование

РП.

3. Проверить сбор передач задач людям, ожидания HITL и доработок. Описать пробелы, способ ежедневного запуска и пересборки. Выбор показателей и согласование ориентиров остаются у РП с владельцем продукта.

Результат работы инженера. План мониторинга технически выполним; источники и ограничения измерений известны до начала Пилота.

POC-22. Рассчитать и согласовать с РП оценочные значения для сравнения

ИИ-инженер — ответственный. Работа и результат.

1. По плану мониторинга поручить агенту рассчитать оценочные значения для выбранного состава работ. Для идеальных часов описать объём выполнения людьми без агентов; для участия людей использовать доступные следы их действий.
2. Сохранить основания: требования, сложность изменений, сопоставимые задачи, доступные записи о проверках и исправлениях. Не считать длительность всей сессии человеческим временем и не выводить трудозатраты людей из токенов.
3. В наборе данных для расчёта показателей сохранить оценки, допущения и ссылки на основания. Организовать проверку в роли РП и записать подтверждение или поправки; в плане мониторинга указать место хранения оценок и согласований. При недостатке данных явно указать ограничения.

Результат работы инженера. Есть согласованные с РП оценочные значения для сравнения, их версии и основания; отсутствие сведений не преобразовано в нулевые затраты.

POC-23. Настроить и проверить ежедневный сбор сведений

ИИ-инженер — ответственный. Работа и результат.

1. Настроить агента внедрения на основе описаний процессов и плана мониторинга: необходимые источники, связи задач и событий, порядок получения и обновления данных, форматы результатов. Источником состава работ служат описанные процессы, а не произвольный сбор всего доступного.
2. Проверить чтение трекера, Git, чата, журналов агентов и сред по применимости. Сохранить исходные ссылки и время событий, отличать первичное событие от его повторного получения, связать MR, прогоны и замечания с задачей.
3. Разделить сбор первички, получение оценок, их подтверждение РП и расчёт метрик. Подготовить набор данных с происхождением фактов и версиями оценок; итоговые показатели должны вычисляться повторяемо.
4. Провести сбор за день, повторить его, затем запросить пропущенные дни или заданный период. Проверить отсутствие дублей и сохранение исправлений; ошибки доступа и пробелы не заменять нулями.
5. Зафиксировать ежедневный автозапуск либо ручной запрос агенту собрать прошлый день. С РП проверить, что запрос за произвольные даты и пересборка выполняются тем же способом, без ручного заполнения таймшитов командой.

Результат работы инженера. Пробный ежедневный сбор и пересборка работают, событие передачи человеку учитывается, источники и правила позволяют проверить полученные данные.

POC-24. Проверить пробную недельную сводку

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Получить замечания РП к пробной недельной сводке и перейти от спорного числа к исходным событиям и оценкам. Проверить связи с задачами, повторные события, границы периода и незавершённую работу.
2. Исправить сбор, правила оценки или вычисление показателя в зависимости от причины. Оценки после изменения вновь предоставить РП; пересчитать затронутый период и сохранить версии.
3. Указать контур и период сводки. Данные временного PoC не называть исходной базой Пилота без проверки сопоставимости; сохранить выявленные ограничения и результат повторной проверки.

Результат работы инженера. В пробной недельной сводке сохранены исправленные значения, причины расхождений и ссылки на использованные данные; существенные ошибки не остались скрыты за итоговыми процентами.

POC-03. Согласовать условия и работы Пилота

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. В паспорте проекта описать свою часть условий Пилота: сопровождение, доступность технической помощи, рабочие версии, инфраструктуру, сбор сведений и ограничения выбранного потока.
2. Указать, какие инженерные действия нужны на первой неделе и какие работы выполняются только при возникновении проблемы. Не подменять задачи продуктовой команды постоянным ведением всего потока инженером.
3. Проверить техническую выполнимость плана и согласовать свой вклад. Условия завершения брать из принятого чек-листа завершения Пилота; само Масштабирование не планировать.

Результат работы инженера. В паспорте проекта записаны условия сопровождения и технической работы, сохранено согласование в части инженера.

POC-04. Подготовить материалы и провести встречу по гейту

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Собрать подтверждения своей части гейта: результаты PoC, рабочую версию, проверки нового контура, маршрут, оценки, пробный сбор и сводку. Для каждого результата дать рабочую ссылку.
2. Показать, где PoC и Пилот отличаются и какие проверки проведены после изменений. Выделить незакрытые условия; запланированное подключение не представлять выполненным.
3. Представить на встрече ограничения и требуемое сопровождение. РП проводит встречу и сохраняет материалы, решение о начале Пилота принимает Спонсор.

Результат работы инженера. Техническая готовность и ограничения представлены для решения по гейту Пилота, область старта соответствует проверенным возможностям.

Пилот

Сопровождать регулярную работу продуктовой команды, обеспечивать достоверные данные и при необходимости улучшать агентов и средства работы. Улучшения и переход к HOTL не обязательны сами по себе; успешность определяется согласованными показателями и условиями завершения.

Чек-лист и работы этапа.

PIL-01. Уточнить критерии и период итоговой оценки

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Проверить измеримость полного набора целевых показателей и пределов баланса из плана мониторинга. Сверить источники, период наблюдения, состав задач, правила сравнения и доступность расчётов.
2. Если критерий невозможно проверить, в плане мониторинга указать недостающие события или данные, влияние на итоговый вывод и предлагаемый способ измерения. Целевые значения согласуются в установленном порядке.
3. При согласованном изменении условий обновить реализацию мониторинга и сохранить версию. Недостигнутый результат не становится успехом только от пересмотра цели: после продолжения требуется следующая проверка.

Результат работы инженера. В плане мониторинга описаны технически проверяемые условия оценки, ограничения и версии метода.

PIL-05. Обеспечить освоение обязанностей участниками Пилота

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

Освоение подтверждают действиями специалиста в своей роли и с учётом фактической помощи. Проверка маршрута относится к PIL-10; отдельное учебное задание для отчёта не требуется.

1. Помогать специалистам продукта при ошибке инструмента, непонятном обращении или затруднении агента. По возможности восстановить инструмент или уточнить навык так, чтобы задачу продолжали агент и ведущий её участник.
2. Не становиться обязательным промежуточным ревьюером всех MR и не вести вместо команды все диалоги. Если пришлось выполнить действие вручную, сохранить его в связи с задачей для учёта помощи.
3. Повторяющиеся технические проблемы записать в журнале улучшений с исходными материалами; организационные — в задачах внедрения и недельной сводке. Однократная помощь не означает передачу инженеру ответственности за весь продуктовый поток.

Результат работы инженера. Команда ведёт задачи сама при доступной инженерной помощи; фактическое участие инженера видно в данных.

PIL-07. Поддерживать рабочую инфраструктуру и доступы

ИИ-инженер — соисполнитель. Ответственный — DevOps. Работа и результат.

1. При изменении платформы, инструментов, моделей, лимитов или прав определить затрагиваемые сценарии. С DevOps сверить изменения с карточкой инфраструктуры и даты их применения.
2. Повторить необходимые проверки после изменения: доступ к данным, выполнение инструмента, сборку и тесты, доставку обращений, чтение журналов мониторингом. Масштаб проверки зависит от затронутой цепочки.
3. Сохранить результаты и связь конфигурации с задачами и периодами. При сбое в задаче устранения сохранить диагностику, затем проверить восстановление и записать результат.

Результат работы инженера. Влияние инфраструктурных изменений проверено и прослеживается в результатах работы агентов.

PIL-09. Актуализировать требования к инфраструктуре и инференсу по результатам Пилота

ИИ-инженер — соисполнитель. Ответственный — DevOps. Работа и результат.

1. По фактическому потоку определить, какие исходные требования пришлось изменить: модель, лимит вызовов, параллелизм, ресурсы платформы, инструменты или среды. В карточке инфраструктуры сохранить ссылки на подтверждения из задач, журналов и измерений.
2. Сверить итоговую конфигурацию и условия инференса с карточкой инфраструктуры. Для меняющихся настроек сохранить дату начала действия и связь с периодами наблюдений.
3. В наборах данных мониторинга сохранить сведения о применённой конфигурации, периодах её действия и расходах. Не приписывать итоговому контуру наблюдения на первоначальном и не оценивать его стоимость по прежним лимитам или тарифным условиям.

Результат работы инженера. В карточке инфраструктуры сохранены уточнённые требования; наборы данных различают конфигурации и позволяют рассчитать показатели для итоговых условий.

PIL-11. Обеспечивать передачу задач от агентов людям

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Проверять, что смена исполнителя с агента на человека сопровождается доступным мониторингу событием. Уточнить с РП различия между передачей, помощью в HITL и обычной приёмкой.
2. Проверить, что после передачи агент прекратил действия, сохранены изменения и человек может продолжить работу. При пропущенном событии исправить его формирование и сохранить исходные подтверждения; пересборка затронутых данных выполняется в PIL-14, а не вторым расчётом в этой работе.
3. По запросу РП разбирать технические затруднения, связанные с передачами. Не требовать от команды объяснения каждой причины и не исключать событие из метрики из-за отсутствия объяснения.

Результат работы инженера. Агент прекратил действия, сохранённая работа доступна человеку; факты передачи пригодны для расчётов в PIL-14–16.

PIL-12. Поддерживать работоспособность обращений HITL

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Если обращения не доходят или ответы не используются агентом, проверить событие, адресата, сообщение, ссылки и чтение ответа. Отличить ошибку доставки от отсутствия реакции участника.
2. Исправить техническую причину и повторить цепочку запроса и продолжения. В задачах внедрения и недельной сводке зафиксировать проблемы доступности людей, замещения и сроков со ссылками на факты ожидания.
3. Повторить затронутый сценарий и подтвердить восстановление обмена: запрос, материалы, ответ и продолжение. Эффект для длительности ожиданий проверяется в PIL-17 по общим данным. При исправной цепочке дополнительные изменения и переход к HOTL не требуются.

Результат работы инженера. Цепочка HITL работает и подтверждена повторной проверкой; нехватка времени или подготовки участников выделена для PIL-05.

PIL-13. Проверять изменения режима человеческого контроля

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Подключаться только при решении проверить изменение контроля в конкретной области. Вместе с РП, разработчиком и тестировщиком указать границы пробы, сохраняемые автоматические проверки, метрики и условия возврата.
2. Настроить режим и сбор наблюдений. Для HOTL проверить, что замечание человека регистрируется асинхронно, связывается с задачей и поступает на исправление; отсутствие блокировки не означает игнорирование замечания.
3. Сравнить показатели до и после с учётом состава задач, поздних дефектов, возвратов и человеческих исправлений. В недельной сводке сохранить сравнение показателей и решение в роли РП; разрешённую область изменения отразить в маршруте задачи. Решение не распространяет режим автоматически на другие типы работ.
4. Если необходимость смены контроля не возникла, отметить неприменимость. Сохраняемый HITL сам по себе не препятствует завершению Пилота.

Результат работы инженера. Для применённого изменения есть измерения и решение, для неприменимой работы — краткое основание без искусственной пробы.

PIL-14. Ежедневно собирать сведения и вычислять метрики

ИИ-инженер — ответственный. Работа и результат.

1. Каждый день обеспечить сбор агентом сведений по описаниям процессов и плану мониторинга. При ручной схеме РП инициирует запрос, инженер поддерживает работоспособность источников и метода.
2. Проверять полноту периода и источников, связывать события задач, MR, запусков, проверок, обращений и передач. Сохранять исходные ссылки и внесённые уточнения, версии конфигурации и периоды её применения.
3. При недоступном источнике явно отметить пробел, устранить причину с DevOps и догрузить день или заданный период. Пересборка должна обновить данные без дублей и не затереть историю согласованных оценок.
4. Готовить данные для оценок и расчётов по сохранённым правилам. Фактическая длительность сессии, время ожидания и человеческое участие — разные величины; отсутствие события не доказывает нулевые затраты.

Результат работы инженера. Первичные сведения доступны с происхождением и полнотой периода; сбор пропущенных дней воспроизводим и не создаёт второй ручной учёт.

PIL-15. Согласовывать оценочные значения с РП

ИИ-инженер — ответственный. Работа и результат.

1. По первичным сведениям рассчитать с агентом объём согласованного состава работ в идеальных часах и оценку человеческого участия. Показать источники, допущения и сомнительные значения; ручной табель команды не вводить.
2. После приёмки уточнить итоговый состав работ. Если оценка изменилась относительно предварительной, сохранить обе версии и записать основания изменения: требования, выясненная сложность или объём.
3. В наборе данных сохранить значения для проверки и согласования, выделить существенные расхождения и аномалии со ссылками на основания. Зафиксировать результат проверки в роли

РП; при поправке сохранить решение и пересчитать затронутые показатели.

4. Включить согласованные оценки в наборы данных с версией и ссылкой на подтверждение. Учесть помощь агенту, ревью, тестирование и исправления в предусмотренных методикой границах.

5. По завершённым задачам выявлять повторяющиеся ошибки оценки объёма, уточнять навык оценивания и проверять новую версию на известных случаях. Согласовать оценочные значения с РП, сохранить версию правил и пересчитать затронутые показатели сопоставимо с исходной базой. При влиянии на предварительные целевые ориентиры записать основание их пересмотра в плане мониторинга; решение о новых условиях не принимать за участников, которым оно принадлежит.

Результат работы инженера. В расчётах используются оценки с основаниями и подтверждением РП, история корректировок не потеряна.

PIL-16. Проверять недельные сводки и полноту данных

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Для замечания к недельной сводке найти использованные версию данных, оценку и правило расчёта. Проверить состав задач, временные границы, незавершённые задачи, повторы и отсутствующие события.
2. Исправлять первичку, оценку и формулу на соответствующем уровне. Изменение оценки сохранить с основаниями и записью проверки в роли РП, изменение метода — с версией; затем пересчитать затронутые периоды.
3. Проверить повторное вычисление по одной версии данных: показатели должны совпадать. Сводка ссылается на подготовленные наборы данных, а не содержит числа, происхождение которых приходится восстанавливать вручную.

Результат работы инженера. В недельной сводке сохранены исправленные сведения и причины расхождения; источник, оценка и метод расчёта прослеживаются раздельно.

PIL-17. Улучшать агентов и способ работы по выявленным ограничениям

ИИ-инженер — ответственный. Работа и результат.

1. Работу запускать при выявленной необходимости. Из замечания или метрики выбрать конкретную проблему, собрать задачу, версию, ожидаемое действие и фактический результат. Сначала определить причину: знания, навык, инструмент, среда или организация работы.
2. Записать ожидаемое изменение и ограниченную область проверки. Сохранить исходную версию, изменить подходящее средство: инструкцию, навык, память, инструмент; проблемы процессов или сред выполнять с ответственными за них участниками.
3. Повторить проблемный сценарий и зависимые проверки. Убедиться, что не исчезла только формулировка ошибки, а изменился результат; проверить побочные последствия по выбранным метрикам.
4. Сохранить запись в журнале улучшений: причина, изменение, версия, проверки и решение. Проверять результат самообучения тем же способом. Если улучшения не нужны, отметить неприменимость без искусственного упражнения.

Результат работы инженера. Нужные изменения имеют доказанный результат либо отменены; объём написанных навыков и число правок не подменяют эффект.

PIL-18. Закрепить проверенные версии и возможность восстановления

ИИ-инженер — ответственный. Работа и результат.

1. Сохранить используемые версии инструкций, навыков, контекста и настроек инструментов в общих системах с доступом участников в их зоне ответственности.
2. Если что-то менялось, связать версии с проверками и периодами наблюдений, сохранить предыдущую рабочую версию и способ восстановления. Не смешивать результаты разных версий без пояснения.
3. Если изменений не было, зафиксировать ту версию, с которой команда действительно работала. Задача не требует специально улучшать агента ради появления новой версии.

Результат работы инженера. Фактическая рабочая версия и материалы доступны, изменения и способ восстановления прослеживаются по сохранённым записям.

PIL-20. Проверить качество, нагрузку людей и стоимость

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Подготовить свою часть полного набора показателей баланса: человеческое участие в помощи, ревью и доработках, расходы на инференс и связь этих затрат с задачами и периодами.
2. С DevOps сверить ресурсы и условия использования итоговой конфигурации. Различить фактические расходы всего Пилота и оценку дальнейшей работы на итоговой инфраструктуре.
3. При пробе HOTL включить последующие исправления и возвраты; не выдавать сокращение блокирующих проверок за снижение затрат, если исправления перенесены на людей позже. В недельной сводке зафиксировать ограничения данных и ссылки на исходные записи.

Результат работы инженера. Сводка учитывает затраты людей и инференса в фактических условиях; экономия не получена за счёт пропуска последующих исправлений.

PIL-19. Подвести количественные итоги по целям внедрения

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Подготовить для РП расчёты по версиям наборов данных и правил: целевые и фактические значения, состав задач, период, объём наблюдений и ограничения сравнения.
2. Связать вывод с итоговой конфигурацией инфраструктуры. Периоды на прежних конфигурациях выделить; при недостатке наблюдений на итоговой указать это ограничение в итогах Пилота, не подменяя фактические сроки пересчётом по предположениям.
3. Отделить факты, оценки и прогнозы, показать затраты внедрения и сопровождения отдельно от изменения производственных затрат. Для спорных чисел предоставить первичные события и согласования оценок.
4. РП фиксирует достижение показателей либо готовит варианты дальнейших действий. Инженер обеспечивает технические доказательства, но не заменяет итоговое решение своей оценкой качества внедрения.

Результат работы инженера. Итоговые значения воспроизводятся, их область и ограничения объяснимы, РП может проверить достижение каждого согласованного показателя.

PIL-06. Закрепить дальнейшее сопровождение и развитие агентов

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. Вместе с РП определить, какие инженерные обязанности сохраняются после Пилота: диагностика агентов, изменение навыков, поддержка мониторинга, работа с замечаниями. Указать конкретного исполнителя, доступность и необходимое время.
2. Проверить его доступ к инструкциям, навыкам, журналам, средствам диагностики и изменениям. Определить общее место замечаний и способ их поступления на разбор.
3. При фактической передаче работать вместе на необходимом действии, затем записать конкретную переданную обязанность и человека. Проба с помощью коллеги ещё не означает передачу ответственности.
4. Не требовать полной передачи всех обязанностей продуктовой команде: сопровождение может продолжаться при согласованных ресурсах. Если улучшения не потребовались, отдельное учебное упражнение для гейта не проводить.

Результат работы инженера. В разделе «Команда проекта» паспорта зафиксированы объём инженерной помощи, назначенный участник, средства и время её выполнения.

PIL-21. Описать проверенный способ работы и границы его применения

ИИ-инженер — ответственный. Работа и результат.

1. Собрать в итогах Пилота ссылки на действующие процессы, рабочие версии агентов, инструкции, примеры задач и сводки. Не копировать их содержимое во второй комплект.
2. Перечислить типы задач, проверки, участие людей, требования к инфраструктуре и сопровождению, при которых получен результат. Указать исключения и ограничения; предположение о переносимости отличать от подтверждённого опыта.
3. С РП и владельцем продукта проверить организационные и продуктовые условия, с разработчиком и тестировщиком — ограничения проверок. Сохранить доступный команде комплект для продолжения работы; новые команды и волны не планировать.

Результат работы инженера. Итоги объясняют, как получен результат и где он применим, материалы доступны без обращения к личной памяти инженера.

PIL-22. Устранить блокирующие отклонения и подтвердить результат

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. По итоговой проверке РП получить конкретные нарушения, относящиеся к агентам или расчётам. Связать каждое с задачей, данными и влиянием на показатель; технические причины среды решать с DevOps.
2. Выполнить необходимые исправления и повторные проверки, получить профильное подтверждение. При изменении данных или версии метода пересчитать затронутые периоды; изменение конфигурации учесть в наблюдениях.
3. Если нарушение не устранено, в задаче исправления и итогах Пилота зафиксировать его состояние и влияние на результат. Решение продолжить Пилот или изменить условия не отмечать как выполненное исправление. При отсутствии нарушений дополнительные работы не требуются.

Результат работы инженера. В задаче исправления и итогах Пилота сохранены подтверждение исправления либо открытое ограничение; согласованные условия не закрыты обещанием будущей доработки.

PIL-02. Проводить ежедневный и недельный разбор внедрения

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. К ежедневному разбору подготовить подтверждённые затруднения: задача, операция, журнал, необходимое решение человека и следующий технический шаг. Отдельно выделить вопросы, которые инженер сам устранить не может.
2. К недельному разбору в недельной сводке записать состояние сбора, исправленные расхождения, изменения агентов и инфраструктуры и результаты проверок. Показать эффект и отсутствие эффекта одинаково явно.
3. Полученные инженерные действия связать с задачами или журналом улучшений и выполнить. Распределение приоритетов и организацию реакции команды обеспечивает РП.

Результат работы инженера. У РП есть факты для решений и состояние инженерных работ, а не дополнительный ручной отчёт участников о каждом часе.

PIL-03. Подвести итоги и зафиксировать статус Пилота

ИИ-инженер — соисполнитель. Ответственный — РП. Работа и результат.

1. В итогах Пилота заполнить свою часть условий завершения: происхождение данных, подтверждения оценок, версии расчётов, актуальную инфраструктуру, ссылки на рабочие материалы и объём сопровождения.
2. Проверить, что показатели рассчитаны для указанной области и условий. Явно отметить неполные данные, незавершённые технические работы и неприменимые условные работы.
3. На итоговом разборе объяснить расчёты и ограничения. При выполнении показателей и обязательных условий РП фиксирует успешное завершение; отдельного технического разрешения инженера или решения Спонсора о принятии не требуется.

Результат работы инженера. В итогах Пилота сохранены проверяемые технические основания итогового статуса и инженерные ограничения; фактический результат отделён от прогноза.

Гайд роли: Аналитик

Обеспечивает проверяемую постановку и уточнение требований при работе команды с автономными агентами. Проверяет понимание задачи, отвечает на предметные вопросы и связывает изменения требований с версиями.

Назначение и распределение работ

Аналитик внедрения может помочь структурировать постановки и вопросы; аналитик продукта подтверждает предметные факты и решения либо получает их у уполномоченного участника. Источник знаний важнее принадлежности к стороне; нельзя выдумать недостающее правило ради завершения постановки.

Конкретные участники, принадлежность к команде и выделенное время — в паспорте проекта. У каждой работы один ответственный; соисполнитель подтверждает свой вклад и не наследует все полномочия ответственного. ID и непосредственные зависимости берутся из чек-листов; порядок чтения ниже не создаёт новых зависимостей. Существующий подходящий результат проверяется и используется по ссылке.

Как выполнять работу роли

Сначала выяснить, как агент понял результат, ограничения и критерии. При работе от замысла проверить подготовленную постановку; при готовом описании проверить его понимание. Ответы и решения сохранять в источнике, доступном агенту, с версией. Повторяющиеся затруднения превращать в проверенное улучшение описания или навыка, а не вручную переписывать каждую задачу.

Результат сохраняется в предусмотренном рабочем документе или событии системы. Отдельный отчёт каждого участника по каждой работе не требуется. На PoC регулярное взаимодействие с агентами ведёт ИИ-инженер, специалисты обучаются и подключаются через него; на Пилоте работают непосредственно.

Подготовка

PREP-28. Подготовить материалы и провести встречу по гейту

Аналитик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Подготовить ссылки на задачи и необходимые исходные материалы к встрече.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Материалы доступны в целевом трекере и соответствуют согласованному объёму.

PREP-15. Описать общий процесс задач

Аналитик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Описать переход от замысла к готовой задаче, возврат на уточнение и продуктовую приёмку.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Для перехода названы условия, сведения и подтверждения.

PREP-06. Провести вводный семинар для участников PoC

Аналитик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. На семинаре разобрать поручение, вопрос агента и способ сохранения ответа.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Запись учебного действия доступна участникам, самостоятельный PoC не требуется.

PREP-30. Подтвердить доступы участников команды

Аналитик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить чтение и изменение требований, вложений и проектного канала своим аккаунтом.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверены именно операции постановки и уточнения.

PREP-03. Выбрать и назначить задачи PoC в целевом трекере

Аналитик — соисполнитель. Ответственный по работе — Владелец продукта. Работа, результат и предшественники.

1. Отобрать доступные задачи по границам паспорта, сохранить ссылки и выделить нерешённые зависимости.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Выбор не требует отдельной карточки методологии и детальной постановки всех будущих задач.

PREP-17. Проверить чтение выбранных задач агентами

Аналитик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Сверить прочитанное агентом описание и вложения с оригиналом.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверено получение содержания, а не только заголовка или номера задачи.

PoC

POC-06. Обучить команду и продемонстрировать работу с агентами

Аналитик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Изучить демонстрацию понимания задачи агентом и предъявления результата; разобрать свои действия.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Есть материалы обучения и ответы на вопросы роли.

POC-18. Адаптировать агентов на задачах PoC

Аналитик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. По вопросам инженера и агентов уточнять факты продукта и причины решений.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Сведения попадают в доступные агенту источники; взаимодействие идёт через инженера.

POC-19. Свести результаты проверок PoC

Аналитик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Подтвердить соответствие проверенных результатов согласованным требованиям.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Непроверенные сценарии и предметные ограничения названы явно.

POC-12. Организовать личную проверку доступов участников

Аналитик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Повторить нужные операции в изменившихся инструментах Пилота.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Доступы временного PoC не приняты за целевые.

РОС-13. Уточнить общий жизненный цикл задач и Git-процесс

Аналитик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Описать уточнения требований и версии постановки в общем ЖЦ.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Новая версия объёма связана с прежней, она не скрывает незавершённую работу.

РОС-14. Описать точки участия человека

Аналитик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Определить вопросы по требованиям, адресатов и содержательный результат ответа.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Понятно, что блокируется и какое событие завершает запрос.

РОС-15. Определить проверки и приёмку результатов

Аналитик — соисполнитель. Ответственный по работе — Тестировщик. Работа, результат и предшественники.

1. Назвать проверяемые критерии, граничные сценарии и различия дефекта и нового требования.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверка связана с версией постановки и нужной ревизией.

РОС-17. Проверить рабочий маршрут в контуре Пилота

Аналитик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. В сквозном прогоне выполнить предусмотренное предметное решение и сохранить ответ.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверен контакт целевого маршрута, а не предположение о доступности аналитика.

Пилот

PIL-02. Проводить ежедневный и недельный разбор внедрения

Аналитик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. На разборе предъявить зависшие вопросы и изменения требований с конкретными задачами.

2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Для следующего шага есть решение или названный вопрос с исполнителем.

PIL-05. Обеспечить освоение обязанностей участниками Пилота

Аналитик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

Освоение подтверждают действиями специалиста в своей роли и с учётом фактической помощи. Проверка маршрута относится к PIL-10; отдельное учебное задание для отчёта не требуется.

1. Перед поручением проверить понимание задачи агентом, затем отвечать на предметные вопросы.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Уточнение смысла не заменяется переписыванием всей постановки вручную.

PIL-10. Вести общий процесс задач и Git

Аналитик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Поддерживать источник, версии и готовность требований в общем ЖЦ.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Чат и трекер не содержат противоречащих действующих постановок.

PIL-12. Поддерживать работоспособность обращений HITL

Аналитик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. На обращениях по требованиям проверить получение вопроса, достаточность материалов и использование ответа агентом; при сбое сохранить конкретный пример для исправления цепочки.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверен полный обмен материалами и продолжение работы; ожидание занятости человека не выдано за техническую ошибку обращения.

PIL-16. Проверять недельные сводки и полноту данных

Аналитик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Сверить выбранные расхождения по объёму и версиям требований с источником.

2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Исправление оценки имеет объяснение изменения состава работ.

PIL-17. Улучшать агентов и способ работы по выявленным ограничениям

Аналитик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Для повторяющихся ошибок понимания уточнить правило описания задач или источник знаний; проверить на примере.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Изменено поведение агента, а не только текст инструкции.

Связь с процессами и источниками

Порядок совместных действий раскрывают гайды управления внедрением, подготовки команды, технической готовности, работы с агентами и управления эффективностью. Показатели и вычисления задаёт план мониторинга; выполненные улучшения — журнал улучшений.

Гайд роли: Архитектор

Подтверждает архитектурные ограничения и решения, существенные для выбранной области агентской разработки. Подключается по необходимости, а не становится обязательным дополнительным гейтом каждой задачи.

Назначение и распределение работ

Роль может выполнять специалист продукта, предприятия или внедрения с достаточным знанием системы. Внешний специалист готовит вариант и доказательства; решение принимается в пределах предоставленных полномочий. Участие в работах ниже условно, если затронуты архитектурные границы или риски.

Конкретные участники, принадлежность к команде и выделенное время — в паспорте проекта. У каждой работы один ответственный; соисполнитель подтверждает свой вклад и не наследует все полномочия ответственного. ID и непосредственные зависимости берутся из чек-листов; порядок чтения ниже не создаёт новых зависимостей. Существующий подходящий результат проверяется и используется по ссылке.

Как выполнять работу роли

Проверять границы компонентов, контрактов, данных и полномочий; отделять реально проверенные свойства от допущений моков и упрощённого PoC. Решение сохранять с причинами, областью и последствиями в существующем архитектурном источнике. Не вводить полный пересмотр архитектуры как условие первого ограниченного внедрения.

Результат сохраняется в предусмотренном рабочем документе или событии системы. Отдельный отчёт каждого участника по каждой работе не требуется. На PoC регулярное взаимодействие с агентами ведёт ИИ-инженер, специалисты обучаются и подключаются через него; на Пилоте работают непосредственно.

Подготовка

PREP-07. Оформить и согласовать карточку инфраструктуры

Архитектор — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. При изменении архитектурных границ указать ограничения размещения, зависимостей и связей.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Допущения карточки инфраструктуры согласуются с устройством продукта.

PREP-30. Подтвердить доступы участников команды

Архитектор — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить доступ к архитектурным материалам и предусмотренным решениям своей роли.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Подтверждены нужные операции без запроса лишних административных прав.

PoC

POC-18. Адаптировать агентов на задачах PoC

Архитектор — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Уточнить причины архитектурных ограничений по вопросам инженера; сохранить решения в доступном источнике.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Агент получает проверенное решение, а не должен догадаться о нём по коду.

POC-07. Уточнить инфраструктуру и доступы для Пилота

Архитектор — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Сопоставить контуры PoC и Пилота по зависимостям, границам и полномочиям.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Упрощённая проба не выдаётся за проверку отсутствовавшей интеграции.

POC-15. Определить проверки и приёмку результатов

Архитектор — соисполнитель. Ответственный по работе — Тестировщик. Работа, результат и предшественники.

1. Определить контрактные и интеграционные проверки затронутого архитектурного решения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверка собственного мока не считается подтверждением реального контракта.

Пилот

PIL-02. Проводить ежедневный и недельный разбор внедрения

Архитектор — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Разбирать архитектурную часть выбранных затруднений на общем ритме.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Новый архитектурный комитет на каждое замечание не создаётся.

PIL-09. Актуализировать требования к инфраструктуре и инференсу по результатам Пилота

Архитектор — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Проверить влияние итоговой инфраструктуры и инференса на принятые границы продукта.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Изменения ограничений отражены в актуальных материалах.

PIL-13. Проверять изменения режима человеческого контроля

Архитектор — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Для рискованной области назвать проверки и ограничения, сохраняемые при пробе контроля.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Снижение HITL не отменяет архитектурные условия совместимости и изоляции.

PIL-16. Проверять недельные сводки и полноту данных

Архитектор — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Уточнить выбранную аномалию, если её причина в архитектурных зависимостях.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Вывод основан на фактах потока, а не общем предположении о сложности.

PIL-17. Улучшать агентов и способ работы по выявленным ограничениям

Архитектор — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. При повторяющейся архитектурной причине выбрать ограниченное изменение и проверить зависимые сценарии.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Рефакторинг не проводится только ради красивой схемы и не объявляется эффектом без проверки.

PIL-21. Описать проверенный способ работы и границы его применения

Архитектор — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Зафиксировать границы архитектурной применимости проверенного способа работы.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Исключения и непроверенные зависимости не скрыты в общей формулировке.

Связь с процессами и источниками

Порядок совместных действий раскрывают гайды управления внедрением, подготовки команды, технической готовности, работы с агентами и управления эффективностью. Показатели и вычисления задаёт план мониторинга; выполненные улучшения — журнал улучшений.

Гайд роли: Разработчик

Обеспечивает правила кодовой базы, ревью, технические проверки и интеграцию. Помогает превратить замечания к агентам в улучшения, проверяемые на реальных изменениях.

Назначение и распределение работ

Разработчик внедрения может адаптировать сборку и проверки, добавлять механики тестирования и разбирать сложные случаи. Ревью и интеграцию выполняет участник с достаточным знанием продукта и правами; принадлежность к внедрению сама по себе их не даёт.

Конкретные участники, принадлежность к команде и выделенное время — в паспорте проекта. У каждой работы один ответственный; соисполнитель подтверждает свой вклад и не наследует все полномочия ответственного. ID и непосредственные зависимости берутся из чек-листов; порядок чтения ниже не создаёт новых зависимостей. Существующий подходящий результат проверяется и используется по ссылке.

Как выполнять работу роли

При ревью проверить смысл изменения, контракт, ошибки, данные и тесты точной ревизии. Замечание формулировать через ожидаемое и фактическое поведение и способ проверки. После исправления проверить результат и затронутую регрессию. Повторяющиеся замечания разбирать с ИИ-инженером, чтобы улучшить навыки и инструменты; не считать ручную доделку нулевой помощью.

Результат сохраняется в предусмотренном рабочем документе или событии системы. Отдельный отчёт каждого участника по каждой работе не требуется. На PoC регулярное взаимодействие с агентами ведёт ИИ-инженер, специалисты обучаются и подключаются через него; на Пилоте работают непосредственно.

Подготовка

PREP-15. Описать общий процесс задач

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Описать переход к выполнению, готовность изменения, возвраты и интеграцию.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Статусы не смешивают публикацию черновика с готовым результатом.

PREP-16. Описать общий Git-процесс

Разработчик — ответственный. Ответственный по работе — Разработчик. Работа, результат и предшественники.

1. Оформить ветки, цели MR, проверки, approvals, конфликты и продвижение ревизии.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Правила одинаковы для людей и агентов, права согласованы с DevOps.

PREP-06. Провести вводный семинар для участников PoC

Разработчик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Разобрать учебный MR, замечание и повторное предъявление на семинаре.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Известен способ сохранить решение и обратную связь доступно агенту.

PREP-13. Проверить технический путь сборки и развёртывания

Разработчик — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Дать точную ревизию, команды сборки и проверок, ожидаемые артефакты; помочь воспроизвести ошибки.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Технический путь проверяется из контура платформы, а не только ноутбука.

PREP-14. Подготовить тестовые среды продукта

Разработчик — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Добавить нужные выбранным сценариям способы управления данными, зависимостями и временем.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Инструменты программно управляемы и не меняют соседние среды.

PREP-30. Подтвердить доступы участников команды

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить чтение и запись рабочей ветки, MR и доступные проверки под своим аккаунтом.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Защищённые ветки и права merge соответствуют назначению.

PoC

POC-06. Обучить команду и продемонстрировать работу с агентами

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Изучить демонстрацию проверки MR и обратной связи через инженера.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Не требуется самостоятельно вести регулярное ревью агентов на PoC.

POC-18. Адаптировать агентов на задачах PoC

Разработчик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Объяснить инженеру причины замечаний по коду; позднее проверять подготовленные им материалы.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Предметная причина сохранена для настройки навыка, а не повторного ручного исправления.

POC-19. Свести результаты проверок PoC

Разработчик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Подтвердить результаты ревью и ограничения проверенных изменений.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Открытые замечания и непроверенные решения видны в журналах.

POC-01. Согласовать область и режимы работы Пилота

Разработчик — соисполнитель. Ответственный по работе — Владелец продукта. Работа, результат и предшественники.

1. По проверенным изменениям подтвердить достаточность инженерных проверок выбранной области.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Непроверенный тип изменений не включён по аналогии с одной удачей.

POC-10. Подготовить тестовые среды для потока Пилота

Разработчик — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Обеспечить сборку и проверки нужной ревизии в целевых средах.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Известны команды и связь образа, SHA и результата теста.

POC-12. Организовать личную проверку доступов участников

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить операции Git и CI в контуре Пилота своей учётной записью.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Права и видимость результатов не выводятся из успешной проверки инженером.

POC-13. Уточнить общий жизненный цикл задач и Git-процесс

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Согласовать ветки, MR, повторное ревью и интеграцию со статусами задач.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Новый commit не наследует проверку старой версии молча.

POC-14. Описать точки участия человека

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Определить точки инженерного HITL, предмет ревью и адресатов.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Запрос начинается с готового комплекта и заканчивается содержательным действием.

POC-15. Определить проверки и приёмку результатов

Разработчик — соисполнитель. Ответственный по работе — Тестировщик. Работа, результат и предшественники.

1. Назвать обязательные проверки кода, контрактов и тестов по изменению.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Зелёный pipeline сам по себе не заменяет продуктовую приёмку.

POC-16. Обеспечить передачу задачи от агента человеку

Разработчик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Проверить получение ветки, MR и контекста при передаче задачи от агента.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Человек может продолжить работу, а агент прекратил конкурирующие изменения.

POC-20. Подготовить и проверить рабочую версию агентов

Разработчик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Повторно проверить устранение своих замечаний в зафиксированной версии.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Известен SHA и набор проверок, а не только новый текст ответа агента.

POC-17. Проверить рабочий маршрут в контуре Пилота

Разработчик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Выполнить ревью и возврат замечания в сквозном прогоне целевого маршрута.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Обратная связь и повторная ревизия доступны агенту.

Пилот

PIL-02. Проводить ежедневный и недельный разбор внедрения

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. На разборе показать повторяющиеся замечания и задержки MR по фактам.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Предложение улучшения относится к группе причин и имеет проверку.

PIL-05. Обеспечить освоение обязанностей участниками Пилота

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

Освоение подтверждают действиями специалиста в своей роли и с учётом фактической помощи. Проверка маршрута относится к PIL-10; отдельное учебное задание для отчёта не требуется.

1. Проводить содержательное ревью по рабочим правилам, проверять код и тесты, возвращать конкретные замечания.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Вердикт относится к ревизии; помощь инженера не снимает роли проверяющего.

PIL-08. Обеспечивать воспроизводимость тестирования

Разработчик — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Обеспечить повторяемую сборку, развёртывание и проверки изменения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Повтор не зависит от несохранённого состояния разработчика.

PIL-10. Вести общий процесс задач и Git

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Сохранять связь задачи с ветками, MR, approvals и интегрированным результатом.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. История позволяет отличить очередь, проверку, исправление и merge.

PIL-11. Обеспечивать передачу задач от агентов людям

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. При передаче принять изменения, сохранить факт ручного продолжения и дальнейший результат.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Человеческий труд после передачи не исчезает из учёта.

PIL-12. Поддерживать работоспособность обращений HITL

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить связь запроса ревью с нужным MR и ревизией, доступность замечаний и вердикта агенту и корректное продолжение после ответа.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверен полный обмен материалами и продолжение работы; ожидание занятости человека не выдано за техническую ошибку обращения.

PIL-13. Проверять изменения режима человеческого контроля

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Определить сохраняемые инженерные проверки и проверить последствия ограниченной пробы контроля.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Отказ от блокирующего ревью не скрывает поздние переделки.

PIL-16. Проверять недельные сводки и полноту данных

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить выбранные расхождения сводки по ревизиям, замечаниям и ручным доработкам.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Исправление данных опирается на Git и результаты проверок.

PIL-17. Улучшать агентов и способ работы по выявленным ограничениям

Разработчик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. С инженером устранить повторяющуюся причину замечаний в навыке, инструменте или тесте; повторить сценарий.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Те же замечания действительно сократились без ослабления требований.

PIL-18. Закрепить проверенные версии и возможность восстановления

Разработчик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Проверить значимые сценарии закрепляемой версии и возможность восстановить проверенный комплект.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Результаты связаны с версиями кода, инструментов и инструкций.

PIL-20. Проверить качество, нагрузку людей и стоимость

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Подтвердить ручные исправления и технические последствия ускорения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Нагрузка ревью и доделок учтена вместе с качеством.

PIL-21. Описать проверенный способ работы и границы его применения

Разработчик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Описать применимые типы изменений и ограничения инженерной проверки.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Документ отражает реальную практику, не обещая пригодность всех репозиториев.

PIL-22. Устранить блокирующие отклонения и подтвердить результат

Разработчик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Повторить проверку исправленного блокирующего отклонения на нужной ревизии.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Закрытие замечания подтверждено поведением и тестами.

Связь с процессами и источниками

Порядок совместных действий раскрывают гайды управления внедрением, подготовки команды, технической готовности, работы с агентами и управления эффективностью. Показатели и вычисления задаёт план мониторинга; выполненные улучшения — журнал улучшений.

Гайд роли: Тестировщик

Обеспечивает достаточные проверки результата, воспроизводимость сценариев и предметные доказательства качества. Помогает развивать тесты, данные и механизмы управления сложными сценариями.

Назначение и распределение работ

Тестировщику внедрения можно поручить автотесты, управляемые моки, время, эмуляторы и диагностику. Тестировщик продукта уточняет значимые сценарии и условия приёмки; конкретное выполнение может распределяться при достаточном знании продукта и доступах.

Конкретные участники, принадлежность к команде и выделенное время — в паспорте проекта. У каждой работы один ответственный; соисполнитель подтверждает свой вклад и не наследует все полномочия ответственного. ID и непосредственные зависимости берутся из чек-листов; порядок чтения ниже не создаёт новых зависимостей. Существующий подходящий результат проверяется и используется по ссылке.

Как выполнять работу роли

Проверять не наличие теста, а нужное поведение и способность обнаружить ошибку. Для дефекта сохранить ревизию, среду, данные, шаги, ожидание и факт. Отделять дефект изменения, сбой среды и новое требование. Использовать изолированные моки, управляемое время и данные там, где они нужны; технические механики описаны в howtos. Не признавать нестабильный тест успешным только по зелёному повтору.

Результат сохраняется в предусмотренном рабочем документе или событии системы. Отдельный отчёт каждого участника по каждой работе не требуется. На PoC регулярное взаимодействие с агентами ведёт ИИ-инженер, специалисты обучаются и подключаются через него; на Пилоте работают непосредственно.

Подготовка

PREP-15. Описать общий процесс задач

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Описать очередь тестирования, начало проверки, возврат и приёмку.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Изменение статуса не названо точным временем начала работы человека.

PREP-06. Провести вводный семинар для участников PoC

Тестировщик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Разобрать учебное предъявление результата и оформление дефекта.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Известно, как сохранить ревизию, шаги и ожидаемый результат.

PREP-14. Подготовить тестовые среды продукта

Тестировщик — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Подготовить повторяемые данные и сценарии, проверить изоляцию и сброс.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Прогон воспроизводится без состояния предыдущей задачи.

PREP-27. Подтвердить готовность систем и связности

Тестировщик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Проверить результаты технического прогона среды и доступность диагностики.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Подтверждена инфраструктура, а не способность агента решить продуктовую задачу.

PREP-30. Подтвердить доступы участников команды

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить доступ к тестам, средам и сообщениям своим аккаунтом.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Можно получить нужную ревизию, результат и диагностические сведения.

PREP-04. Согласовать сроки и работы PoC

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Уточнить направления проверок будущих проб по границам продукта.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Способ проверки обеспечивается до зависимой задачи, не требуется полный набор заранее.

PoC

POC-06. Обучить команду и продемонстрировать работу с агентами

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Изучить демонстрацию проверок инженером и порядок предъявления замечаний.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. На PoC обязательны обучение и демонстрация, не самостоятельное ведение потока.

POC-18. Адаптировать агентов на задачах PoC

Тестировщик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. По запросам инженера уточнять тестовые сценарии и причины дефектов; участвовать в поздних пробах через него.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Агенту возвращается проверенная обратная связь, без тупикового общения.

POC-19. Свести результаты проверок PoC

Тестировщик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Подтвердить качество проверок, найденные дефекты и непроверенные свойства.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Журналы содержат ревизии и доказательства, а не только число зелёных запусков.

POC-01. Согласовать область и режимы работы Пилота

Тестировщик — соисполнитель. Ответственный по работе — Владелец продукта. Работа, результат и предшественники.

1. Подтвердить проверяемость включённых типов работ и назвать ограничения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Непокрытая область не объявлена проверенной по соседнему сценарию.

РОС-10. Подготовить тестовые среды для потока Пилота

Тестировщик — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Задать нужные сценариям данные и подтвердить e2e на целевой среде.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверяются значимые взаимодействия, а не только доступность URL.

РОС-12. Организовать личную проверку доступов участников

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Лично выполнить нужные проверки доступа в новых системах Пилота.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Участник может получить результаты и передать замечание по целевой схеме.

РОС-14. Описать точки участия человека

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Определить предмет HITL тестирования, готовый комплект и проверяющих.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Ожидание среды отделено от очереди тестировщика.

РОС-15. Определить проверки и приёмку результатов

Тестировщик — ответственный. Ответственный по работе — Тестировщик. Работа, результат и предшественники.

1. Заполнить порядок приёмки в ЖЦ: проверки, доказательства, дефекты, повтор и решение владельца продукта.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Для каждого типа работ названы результат и способ его подтвердить.

РОС-20. Подготовить и проверить рабочую версию агентов

Тестировщик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Повторить проверки после исправлений агента и подтвердить свои замечания.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Зафиксированная версия пригодна в согласованной области.

РОС-17. Проверить рабочий маршрут в контуре Пилота

Тестировщик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Проверить предъявление и исправление результата в целевом сквозном маршруте.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Дефект, сбой среды и новое требование различаются.

Пилот

PIL-02. Проводить ежедневный и недельный разбор внедрения

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Предъявлять группы возвратов, нестабильных тестов и ожиданий с примерами.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Выбрано действие по причине, а не требование бесконечно увеличивать число тестов.

PIL-05. Обеспечить освоение обязанностей участниками Пилота

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

Освоение подтверждают действиями специалиста в своей роли и с учётом фактической помощи. Проверка маршрута относится к PIL-10; отдельное учебное задание для отчёта не требуется.

1. Проверять результаты непосредственно по критериям и возвращать воспроизводимые замечания.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Принятие связано с ревизией и данными, инженер помогает при затруднении.

PIL-08. Обеспечивать воспроизводимость тестирования

Тестировщик — соисполнитель. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Повторять проверки из известного состояния, разбирать нестабильность и неполноту.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Повтор до первого зелёного результата не считается доказательством качества.

PIL-10. Вести общий процесс задач и Git

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Сохранять начало/результат проверки, причину возврата и связь с новой ревизией.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. История позволяет восстановить раунды и ожидания.

PIL-12. Поддерживать работоспособность обращений HITL

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить получение запроса на проверку, наличие ревизии и результатов, сохранение замечаний и их получение агентом для исправления.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверен полный обмен материалами и продолжение работы; ожидание занятости человека не выдано за техническую ошибку обращения.

PIL-13. Проверять изменения режима человеческого контроля

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Определить автоматические замены и наблюдение за возвратами перед изменением ручного контроля.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Поздние дефекты и HOTL-исправления учитываются после введения режима.

PIL-16. Проверять недельные сводки и полноту данных

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Уточнить выбранные расхождения по дефектам, возвратам и результатам тестирования.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Недоступный прогон не объявлен успешным.

PIL-17. Улучшать агентов и способ работы по выявленным ограничениям

Тестировщик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Сохранить повторяющуюся проверку как поддерживаемый тест или улучшить данные и инструмент; проверить обнаружение ошибки.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверено поведение, а не только наличие нового тестового файла.

PIL-18. Закрепить проверенные версии и возможность восстановления

Тестировщик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Повторить значимые сценарии закрепляемой версии и указать остаточные ограничения.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Доказательства соответствуют точному комплекту версий.

PIL-20. Проверить качество, нагрузку людей и стоимость

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Подтвердить дефекты, возвраты, часы проверок и качество в итоговом окне.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Баланс не подменён общим числом найденных замечаний.

PIL-21. Описать проверенный способ работы и границы его применения

Тестировщик — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Назвать проверенные сценарии, обязательный ручной контроль и ограничения данных.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Способ работы не распространяется за пределы подтверждённого покрытия.

PIL-22. Устранить блокирующие отклонения и подтвердить результат

Тестировщик — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Повторить сценарий блокирующего отклонения после исправления и затронутую регрессию.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Исправление подтверждено и не скрывает новый дефект.

Связь с процессами и источниками

Порядок совместных действий раскрывают гайды управления внедрением, подготовки команды, технической готовности, работы с агентами и управления эффективностью. Показатели и вычисления задаёт план мониторинга; выполненные улучшения — журнал улучшений.

Гайд роли: DevOps

Обеспечивает ресурсы, подключение и эксплуатацию платформы, конвейера и тестовых сред. Подтверждает работоспособность из нужного контура и сохранение фактической конфигурации.

Назначение и распределение работ

DevOps внедрения может установить платформу и временные инструменты PoC; DevOps продукта или предприятия обеспечивает ресурсы и права своей инфраструктуры. Если предоставление и установка выполняются разными людьми, работа разделяется в проектном трекере с одним ответственным за каждую часть. Дополнительные роли администраторов в проект не вводятся.

Конкретные участники, принадлежность к команде и выделенное время — в паспорте проекта. У каждой работы один ответственный; соисполнитель подтверждает свой вклад и не наследует все полномочия ответственного. ID и непосредственные зависимости берутся из чек-листов; порядок чтения ниже не создаёт новых зависимостей. Существующий подходящий результат проверяется и используется по ссылке.

Как выполнять работу роли

Выполнять только назначенные операции в пределах полномочий. Проверять связь задача → SHA → артефакт/digest → деплой → среда → тест. Сохранять результат выдачи и проверки прав, а не секреты. Различать выделение ресурса, готовность среды, очередь CI и отказ. После изменения повторять затронутые проверки; диагностика доступности не заменяет продуктовую приёмку.

Результат сохраняется в предусмотренном рабочем документе или событии системы. Отдельный отчёт каждого участника по каждой работе не требуется. На PoC регулярное взаимодействие с агентами ведёт ИИ-инженер, специалисты обучаются и подключаются через него; на Пилоте работают непосредственно.

Подготовка

PREP-28. Подготовить материалы и провести встречу по гейту

DevOps — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Представить фактические проверки систем, связности, доступов и сред.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Заявки на ресурсы не выданы за готовую инфраструктуру.

PREP-16. Описать общий Git-процесс

DevOps — соисполнитель. Ответственный по работе — Разработчик. Работа, результат и предшественники.

1. Описать CI, protected-ветки, runner, registry и ограничения операций.

2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Разработчик может связать эти действия с общими правилами Git.

PREP-07. Оформить и согласовать карточку инфраструктуры

DevOps — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Описать размещение, сети, ресурсы, аккаунты и разрешённые операции людей и платформы.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Все системы сведены в единую карточку, распределены предоставление и установка.

PREP-08. Развернуть или подключить платформу

DevOps — ответственный. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. На предоставленных ресурсах установить платформу либо подключить существующую; проверить состояние и диагностику.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. On-prem установка не начинается без ресурсов и доступа; облако не требует лишней локальной копии.

PREP-09. Подключить и проверить инференс

DevOps — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Обеспечить разрешённый сетевой маршрут и аутентификацию инференса.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверен запрос из контура платформы и обработка ошибки, а не только ping.

PREP-11. Настроить связи и доступы системам и ИИ-платформе

DevOps — ответственный. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Настроить связи, служебные аккаунты и операции интеграций из карточки.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Личные учётные записи не используются как скрытая замена системным подключениям.

PREP-12. Подготовить рабочие места и доступы людям

DevOps — ответственный. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Выдать доступы по списку ролей, настроить окружение и безопасное получение секретов.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Аккаунты и права соответствуют нужным операциям; секретов в документах нет.

PREP-13. Проверить технический путь сборки и развёртывания

DevOps — ответственный. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Провести точную ревизию через сборку, хранилище и деплой; сохранить цепочку идентификаторов.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Фактически запущенный артефакт совпадает с проверенным.

PREP-14. Подготовить тестовые среды продукта

DevOps — ответственный. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Подготовить среды, изоляцию, сброс и программное управление зависимостями с профильными специалистами.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Параллельная работа не меняет чужие данные и результаты.

PREP-27. Подтвердить готовность систем и связности

DevOps — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Устранить проблемы систем и сетей, повторить затронутые технические проверки.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Подтверждения относятся к актуальной конфигурации и служебным аккаунтам.

PREP-30. Подтвердить доступы участников команды

DevOps — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Устранить обнаруженные участниками ошибки прав и обеспечить повторную личную проверку.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Проверка не заменена входом DevOps с более широкими правами.

PoC

POC-07. Уточнить инфраструктуру и доступы для Пилота

DevOps — ответственный. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Сопоставить временную и целевую инфраструктуру, обновить необходимые подключения и ресурсы.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Все различия перед Пилотом отражены в карточке с нужными проверками.

POC-08. Подготовить системы и ресурсы Пилота

DevOps — ответственный. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Развернуть или подключить недостающие системы Пилота и проверить функции.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Существование namespace или сервиса само по себе не подтверждает работоспособность.

POC-09. Настроить связность и выдать доступы

DevOps — ответственный. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Настроить целевые сетевые маршруты и права по подтверждённым операциям.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Подключения проверены в нужном направлении под реальными аккаунтами.

POC-10. Подготовить тестовые среды для потока Пилота

DevOps — ответственный. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Подготовить среды для потока Пилота, изоляцию, данные, диагностику и резервирование.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Целевые сценарии выполняются на нужных ревизиях без перезаписи чужой приёмки.

POC-11. Проверить техническую готовность из контура Пилота

DevOps — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Исправить технические ошибки сквозной проверки и повторить зависимые шаги.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Частичная недоступность не скрыта общим успешным статусом.

POC-12. Организовать личную проверку доступов участников

DevOps — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Проверить собственные операции сопровождения и помочь с ошибками прав других участников.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Есть личные подтверждения доступов Пилота.

POC-13. Уточнить общий жизненный цикл задач и Git-процесс

DevOps — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Согласовать автоматические события CI и деплоя с состояниями задач и MR.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Kubernetes API управляет ресурсами, а readiness приложения подтверждается отдельной проверкой.

POC-23. Настроить и проверить ежедневный сбор сведений

DevOps — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Обеспечить чтение нужных журналов, API и истории для сборщика.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Источники доступны по разрешённым операциям, пробелы отмечены.

Пилот

PIL-02. Проводить ежедневный и недельный разбор внедрения

DevOps — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. На разборе предъявлять очередь сред, отказов и ресурсных ограничений с фактами.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Предложение изменения связано с задержкой задач, а не только технической загрузкой.

PIL-06. Закрепить дальнейшее сопровождение и развитие агентов

DevOps — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Подтвердить ресурс, замещения и доступы для дальнейшего сопровождения платформы и сред.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. После Пилота инфраструктура не остаётся без ответственного.

PIL-07. Поддерживать рабочую инфраструктуру и доступы

DevOps — ответственный. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Поддерживать доступность, реагировать на ошибки и проверять восстановление затронутых операций.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Время общего инцидента не приписывается незатронутым задачам.

PIL-08. Обеспечивать воспроизводимость тестирования

DevOps — ответственный. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. Обеспечить воспроизводимый деплой, данные, сброс и диагностику тестирования.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Версия и состояние приёмки защищены от конкурирующего pipeline.

PIL-09. Актуализировать требования к инфраструктуре и инференсу по результатам Пилота

DevOps — ответственный. Ответственный по работе — DevOps. Работа, результат и предшественники.

1. По фактической нагрузке актуализировать мощности, лимиты, инференс и условия эксплуатации.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. В карточке сохранены изменения и итоговые требования, метрики связаны с ними.

PIL-14. Ежедневно собирать сведения и вычислять метрики

DevOps — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Восстанавливать техническую доступность источников сборщика и сохранять причины неполноты.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Пропуски догружаются после восстановления без двойного счёта.

PIL-16. Проверять недельные сводки и полноту данных

DevOps — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Сверять инфраструктурные расхождения сводки по логам, конфигурациям и периодам.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Очередь, исполнение и отказ различаются.

PIL-17. Улучшать агентов и способ работы по выявленным ограничениям

DevOps — соисполнитель. Ответственный по работе — ИИ-инженер. Работа, результат и предшественники.

1. Выполнять выбранные изменения сред и инструментов с проверкой проблемного сценария.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Журнал содержит фактического исполнителя, версию и эффект.

PIL-20. Проверить качество, нагрузку людей и стоимость

DevOps — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Подтвердить фактические расходы и ограничения инфраструктуры принятого периода.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Общие расходы не учтены повторно в каждом продукте.

PIL-22. Устранить блокирующие отклонения и подтвердить результат

DevOps — соисполнитель. Ответственный по работе — РП. Работа, результат и предшественники.

1. Устранить блокирующую инфраструктурную причину и повторить затронутые проверки.
2. Связать выполненное действие и его подтверждение с рабочим источником результата, не создавая вторую копию уже сохранённых сведений.

Проверка своего вклада. Восстановление подтверждено операцией задачи, а не только поднятым сервисом.

Связь с процессами и источниками

Порядок совместных действий раскрывают гайды управления внедрением, подготовки команды, технической готовности, работы с агентами и управления эффективностью. Показатели и вычисления задаёт план мониторинга; выполненные улучшения — журнал улучшений.

Практические материалы

Общие способы организации внедрения и правила расчётов, которые используются на разных этапах и в работе нескольких ролей. Эти материалы дополняют гайды и шаблоны.

- Метрики и правила расчёта: показатели результата, процесса и баланса на уровне команды и предприятия, исходные данные, формулы и ограничения выводов.
- Словарь методологии: основные понятия, ИИ-платформа, участие людей и уровни внедрения.
- Офис внедрения ИИ: единые требования, корпоративные ограничения, инфраструктура и сопоставимые результаты программы.
- Автоматический сбор сведений и расчёт метрик: извлечение фактов из рабочих систем и свободного текста, оценки, происхождение данных, ежедневный сбор и пересборка.
- Проектирование тестовой инфраструктуры: независимые среды, развёртывание ревизий, моки, эмуляторы, управление временем, данными и проверками.
- Стратегия тестирования: развитие повторяемых проверок, доказательства и улучшения тестового набора.
- Агент и репозиторий внедрения: рабочие документы, настройки и права агента, запуск помощи по методологии и ведение журналов.

Офис внедрения ИИ

Офис обеспечивает единые условия внедрения агентской разработки на предприятии: применение корпоративных политик и ограничений, допустимые модели и способы подключения инференса, порядок получения инфраструктуры, обязательные метрики и сопоставимую отчётность. Он помогает избежать дублирования инфраструктуры и расходов, сохраняя необходимые различия продуктов. Методологическая помощь, обучение и разбор сложных случаев поддерживают эту работу.

Офис — организационная функция, а не роль одного специалиста. Его устройство определяется программой. Роли руководителя программы и методолога офиса внедрения не подменяют друг друга: первый координирует программу, второй ведёт общие правила и проверяет их применение.

Когда организовать офис

На этапе прогнозирования эффектов отдельный офис не требуется. Ограниченные пробы позволяют получить основания для решения о программе без предварительного создания полной структуры.

В пилотной волне общие функции может выполнять назначенный методолог вместе с руководителем программы. При подключении многих команд нужен постоянный ресурс для единых правил, инфраструктурных потребностей и сведения результатов. Более десяти команд — организационный ориентир для отдельного офиса, а не формальный порог: разнообразие продуктов и число одновременных внедрений также влияют на нагрузку.

Внедрение планируется на уровне предприятия: прогнозирование эффектов → пилотная волна → волны масштабирования → подведение итогов. Офис не превращает успешный продуктовый Пилот в отдельный процесс распространения на всё предприятие. Состав продуктов, волны, ресурсы и полномочия задаются паспортом программы; каждый продукт проходит свои этапы и гейты.

Общие требования и решения

Направление	Что закрепляется и проверяется
Модели и инференс	Допустимые модели, варианты подключения, границы передачи данных и порядок согласования исключений в рамках корпоративных политик
Инфраструктура	Как запрашиваются ресурсы, какие решения переиспользуются, кто обеспечивает подключение и сопровождение; как учитываются общие мощности и расходы
Процессы и контроль	Обязательные требования к общему процессу людей и агентов, проверкам и полномочиям; допустимые адаптации продукта
Метрики и отчётность	Единые определения, обязательный состав сведений, требования к происхождению оценок, версиям и полноте; сопоставимые границы затрат
Компетенции и практики	Материалы обучения, помощь в применении правил, проверенные способы настройки и разбора повторяющихся проблем

Методолог готовит и поддерживает требования, согласует их с уполномоченными участниками и проверяет применение. Он не создаёт в рамках внедрения новые корпоративные политики ИБ и не выдаёт себе право менять доступы. Вопросы, требующие изменения корпоративных правил, решаются через существующий порядок предприятия; программа учитывает принятое решение или действует в текущих ограничениях.

Для требования сохраняются версия, область применения, основание, согласование и допустимые исключения. Исключение содержит продукт, причину, ограничения, срок пересмотра и решение уполномоченного участника. Различия стеков и бэклогов не являются нарушением сами по себе; единообразие требуется там, где оно обеспечивает управляемость, сравнимость и общие расходы.

Участники и полномочия

Состав офиса, функции, конкретные участники и выделенное время фиксируются в паспорте программы. Руководитель программы организует общие зависимости и материалы решений; методолог — требования, методическую поддержку и сопоставимость. ИИ-инженеры и DevOps подключаются к общим техническим работам по назначениям. Не требуется создавать новую проектную роль под каждую функцию офиса.

Комитет внедрения собирает участников, уполномоченных решать вопросы повестки: Спонсора или руководителя программы, методолога, владельцев и архитекторов затронутых продуктов, участников общих технических работ. Постоянный состав и границы решений задаются программой. Комитет рассматривает варианты, а решения о ресурсах, приоритетах и исключениях принимаются в пределах полномочий конкретных участников и сохраняются в протоколе.

РП продуктов отвечают за участие команд и достоверность локальных сведений; владельцы продуктов — за продуктовые решения. Офис не выполняет вместо них все действия внедрения. Порядок участия специалистов команды внедрения определяется конфигурацией и назначениями.

Ритм работы

Формат	Начальный ориентир	Результат
Рабочая сверка	Еженедельно: руководитель программы, методолог и участники затронутых внедрений	Общие препятствия, потребности в инфраструктуре и помощи, конкретные действия с ответственными и сроками
Комитет	Ежемесячно и перед существенными решениями по программе	Решения по приоритетам, ресурсам, исключениям и следующим волнам в пределах утверждённых полномочий
Разбор практики и обучение	По потребности, например ежемесячно	Проверенные примеры, обновления материалов и действия по повторяющимся проблемам

Точный календарь определяется программой, с использованием действующих встреч предприятия. Срочный блокер передаётся уполномоченному участнику сразу; ближайшее заседание не должно быть единственным способом его решения. Общий канал или очередь обращений обеспечивает назначение исполнителя и видимость состояния вопроса.

К встрече готовятся сводка, расхождения и конкретные варианты решения. В протоколе сохраняются основания, принятое решение, область действия, ответственный, срок и следующий контрольный момент. Локальные работы остаются в трекере продукта; общие зависимости — на уровне программы. Дневной сбор данных и недельные разборы продуктов идут по процессу управления эффективностью.

Как сводить результаты продуктов

1. Получить версии планов мониторинга и журналов показателей, период наблюдения, область задач, исходы, конфигурацию и ограничения каждого продукта.
2. Проверить одинаковый смысл единиц и границ: начало/конец поставки, оценка объёма и человеческого труда, режим контроля, состав проверок, полнота и зрелость наблюдений. Неподтверждённые данные сохранить с ограничениями и назначить уточнение.
3. Объединять только сопоставимые данные. Медианы пересчитывать по исходным наблюдениям, доли — по общим числителю и знаменателю. Проценты улучшений разных продуктов не усреднять как эффект предприятия; несопоставимые группы показывать отдельно.
4. Показывать число и долю команд, завершивших PoC и перешедших к Пилоту, а также число и долю успешно завершивших Пилот. Основания — действующие чек-листы и записи итогов, а не общая формулировка «регулярно применяют агентов».
5. Отделять полученные результаты от прогнозов, локальные затраты от расходов офиса и общих сервисов. Не учитывать общие расходы повторно в каждой команде. Правила — в метриках предприятия.
6. Выделить общие препятствия и предложить действия: изменение доступного ресурса, общего инструмента, правил либо поддержки. Проверить результат на следующем разборе.

Сбор и сведение могут выполняться агентом по настроенному навыку с сохранением источников и версий. Согласованные оценки продуктов не переписываются молча ради удобства общей отчётности. Пробелы источников остаются видимыми.

Как организовать работу

На пилотной волне назначьте исполнителей общих функций и выделенное время; запишите минимальные требования к моделям, получению инфраструктуры, показателям и исключениям. Проверьте, что команды могут использовать эти требования в паспортах, инфраструктурных карточках и планах мониторинга.

Перед расширением волн оцените реальную нагрузку на сопровождение и согласование. Уточните состав офиса и комитета, календарь и место общих материалов. Проверьте сведение первых доступных результатов без потери различий продуктов. Не считайте офис работающим только по факту назначения руководителя: должны выполняться согласованные функции, приниматься решения и проверяться их исполнение.

При подведении итогов закрепите действующие требования, исключения, доступные сервисы и дальнейшую поддержку. Версии материалов, навыков и учебных примеров сохраняются в доступном командам хранилище; их применимость проверяется при использовании в другом продукте.

Словарь методологии

Краткое руководство · Процесс внедрения · Гайды и шаблоны

Основные понятия

Термин	Значение в методологии
Внедрение ИИ в разработку ПО	Организация устойчивого применения ИИ в работе предприятия и команд, включая изменения в компетенциях людей, процессах и технологиях. Основной акцент методологии — на агентской разработке.
Агентская разработка ПО	Разработка ПО с использованием ИИ-агентов: участники команды передают агентам выполнение задач, определяют ограничения и критерии приёмки и проверяют результаты в установленном порядке. Охват задач и автономность расширяются по подтверждённым результатам.
Агент	Программная система на базе ИИ, которая выполняет поставленную задачу: планирует действия, использует инструменты, оценивает промежуточные результаты и корректирует дальнейшие шаги в пределах заданных полномочий.
Автономность агента	Степень самостоятельности агента: какие действия он может выполнять без отдельного согласования и где требуется участие человека. Автономность задаётся для конкретной области работ и не означает отсутствие контроля.
Агент внедрения	Агент, который объясняет методологию, показывает недостающие результаты и возможные следующие шаги, собирает сведения, рассчитывает объём и затраченное время, вычисляет метрики и готовит отчётность. Он не принимает решения о целях и плане вместо участников проекта.
Производство ПО	Техническое проектирование, написание и проверка кода, тестирование, исправление дефектов и подготовка выпуска на основе согласованных требований. Бизнес-анализ и эксплуатация в этот контур не входят.
Трудозатраты на производство ПО	Затраты рабочего времени людей на работы, входящие в производство ПО. Состав учитываемых работ фиксируется для выбранного потока задач. Время агентов, токены и стоимость платформы не переводятся в человеко-часы. Правила оценки — в правилах расчёта.

В тексте **«агент» означает ИИ-агента** и пишется со строчной буквы, кроме начала предложения и заголовка. Другие виды агентов называются с уточнением: агент мониторинга, агент сборки и т. п. В самостоятельном документе при первом упоминании допустимо «ИИ-агент (далее — агент)».

«Внедрение ИИ в разработку ПО» используется для программы изменений; «агентская разработка» — для подхода; «работа с агентом» и «применение агентов» — для конкретных действий. Словосочетание «разработка агентов» означает создание самих агентов и не заменяет название

подхода.

Участие человека

Термин	Значение в методологии
HITL — human-in-the-loop	Человек проверяет или принимает решение до зависимого перехода; без ответа этот переход ждёт.
HOTL — human-on-the-loop	Человек контролирует асинхронно: переход не ждёт вердикта, находки требуют последующей обработки и могут привести к остановке или возврату режима. Human-out-of-the-loop означает отсутствие такого оперативного участия.
Без плановой ручной проверки	Для выбранной работы нет обязательного просмотра каждой задачи или выборки; автоматические проверки, ограничения и обращения по исключениям сохраняются.

Состав агентских работ и режим проверки задаются независимо; подробности и пример.

Технологии

Термин	Значение в методологии
ИИ-платформа, также **платформа выполнения агентов**, кратко **платформа**	Среда, обеспечивающая непрерывную работу автономных агентов: запуск задач по запросам, событиям и расписанию, управление выполнением, подключение инструментов и проектных систем, хранение состояния и результатов, наблюдение и контроль. Включает управление очередями и параллельными запусками, права инструментов, взаимодействие с людьми и сохранение истории. Инференс может предоставляться отдельным сервисом. Конфигурация конкретного проекта описывается в карточке инфраструктуры.
Модель	Компонент ИИ, используемый агентом для обработки контекста, формирования ответов и выбора действий. Модель сама по себе не заменяет агента, его инструменты и среду выполнения.
Инференс	Выполнение запросов к модели; сервис инференса предоставляет агентам такую возможность через согласованный интерфейс. Размещение инференса и платформы выбирается независимо.
Знания и инструкции агента, также **литература проекта**	Поддерживаемые сведения о продукте и правила работы: требования, архитектурные описания, промпты, навыки, инструкции и заметки с определёнными источниками и версиями.
Навык агента	Переиспользуемая инструкция и связанные средства для выполнения определённого вида работы. Состав и способ подключения зависят от выбранного инструментария.

Уровни и участники внедрения

Термин	Значение в методологии
Предприятие	Уровень программы: прогнозирование эффектов, пилотная волна, волны масштабирования и подведение итогов. Общие требования, выбор продуктов, ресурсы и сведение результатов задаются на этом уровне.
Команда	Уровень применения практик к конкретному продукту и потоку задач. Каждая подключаемая команда проходит Подготовку, PoC и Пилот.
Пилотная команда	Команда, выбранная для первых внедрений программы. Не тождественна этапу «Пилот», который проходят и последующие команды.
Команда внедрения	Выделенные специалисты, помогающие внедрить агентскую разработку и освоить работу. Это может быть подготовленная команда предприятия, профессиональная внешняя команда или их сочетание. Не подменяет участие и решения продуктовой команды.
РП	Роль организации работ внедрения. РП команды внедрения организует её работы, РП продукта — участие продуктовой команды; конкретные назначения и совмещения задаются в паспорте. Гайд роли.
ИИ-инженер	Развивает агентов, инструкции, навыки, контекст и инструменты, разбирает затруднения, проверяет улучшения и обеспечивает сбор сведений. Может назначаться из команды продукта или внедрения. Гайд роли.

PPT — **люди, процессы, технологии** — определяет, что меняется. Предприятие и команда определяют, где выполняются работы и принимаются решения. Этапы определяют, когда нужны соответствующие результаты. Эти три измерения применяются совместно.

Агент и репозиторий внедрения

Агент внедрения помогает применять методологию к продукту или программе: объясняет шаги и развилки, показывает недостающие результаты, собирает сведения, вычисляет метрики и готовит отчётность. Участники задают цели, принимают решения и выполняют работы своих ролей. Агенту нужен доступ к действующим материалам и подтверждениям, чтобы рекомендации опирались на состояние проекта.

Репозиторий внедрения хранит рабочие документы, адаптации и версии настроек агента. Это может быть отдельный Git-репозиторий или раздел уже существующего репозитория. Адрес и организацию хранения определяют в проекте; установка ИИ-платформы сама по себе их не выбирает. Исходники продукта остаются в продуктовых репозиториях.

1. Подготовить место для материалов

На Подготовке РП организует доступное участникам место, ИИ-инженер создаёт структуру и проверяет работу с ней. Если для Git нужны подключения и учётные записи, они включаются в карточку инфраструктуры. Конкретные ответственные назначаются в работах проекта.

Зафиксируйте сервер и репозиторий, ветку, корневой каталог и порядок изменения документов. Сохраните ссылку на исходную методологию и её неизменяемый commit/tag. При файловой передаче укажите версию пакета и доступное хранилище. Ссылка на постоянно меняющуюся папку не позволяет восстановить использованные правила.

В README репозитория укажите продукт/программу, источник методологии, действующие документы, место инструкций агента и доступные каналы общения. Используйте относительные ссылки на документы этого репозитория и проверяемые адреса внешних рабочих систем.

Пример структуры, которую можно адаптировать к существующему размещению:

```
README.md       # навигация, версия методологии и порядок обновления
project/        # паспорт, процессы, инфраструктура, план мониторинга
checks/         # рабочие чек-листы и ссылки на подтверждения
logs/           # дневные записи, показатели и улучшения
agents/         # инструкции, навыки и настройки агента или ссылки на них
```

Это пример каталогов, не обязательный комплект копирования. Создавайте документы по мере выполнения работ. Состав команды и сроки ведутся в паспорте, работы — в чек-листах и трекере. Если подходящий документ уже есть, подключите его ссылкой вместо создания второй действующей копии. Примеры методологии не копируются как факты проекта.

2. Зафиксировать рабочие сведения

Начните с паспорта продукта либо паспорта программы. Для продукта подключите чек-лист Подготовки, карточку инфраструктуры и фактические описания ЖЦ задач и Git-процесса. На Подготовке цель содержательная; план мониторинга с метриками заполняется на PoC по этапному порядку.

Сведения хранятся там, где ими управляет команда: требования и задачи — в трекере, обсуждения — в рабочем чате, изменения — в Git. Репозиторий внедрения связывает их и хранит проектные правила. Не требуется заранее собрать полный индекс знаний о продукте: содержание уточняется на пробах PoC.

3. Подключить агента

До готовности ИИ-платформы документы может помогать вести доступный инженеру локальный или облачный агент в пределах разрешённых подключений. При установке платформы on-prem либо использовании облачного варианта ему предоставляют те же действующие материалы и согласованные права. Способ развёртывания, инференс, передача данных и подключения берутся из карточки инфраструктуры, а конкретная установка выполняется по документации выбранной платформы.

В инструкциях агента зафиксируйте:

- источник и версию методологии, репозиторий и ветку рабочих документов;
- продукт/программу, этап, назначения ролей и полномочия на изменения;
- источники требований, решений и подтверждений, доступные каналы вопросов;
- правило различать планы, оценки и подтверждённые факты; не заполнять неизвестное вымыслом;
- порядок предложения изменений документов и их принятия в Git;
- на PoC — правила сбора сведений по плану мониторинга и места журналов.

Проверьте чтение документов, переход по ссылкам, доступ к предусмотренным коммуникациям и сохранение изменения в разрешённой ветке. Проверка инженером под личной учётной записью не подтверждает права агента. Секреты хранятся штатным способом, в документах остаются идентификаторы и подтверждения доступов.

4. Начать работу с агентом

Первое поручение может выглядеть так:

> Прочитай методологию указанной версии и документы проекта в этом репозитории. Покажи, какие результаты текущего этапа подтверждены, каких не хватает и к каким работам можно приступить. Для каждого вывода дай ссылку на основание. Отдельно перечисли неизвестные сведения; не считай заполненную форму доказательством выполнения.

РП и ИИ-инженер уточняют неверные выводы, доступы и назначения. Агент предлагает ближайшие действия на основе гейта и работ, а не создаёт новый независимый план поверх согласованных материалов. Решение о приоритетах и запуске остаётся у уполномоченных участников.

После работы участник сообщает агенту результат или предоставляет ссылку на запись системы. Агент связывает подтверждение с работой и обновляет документ по согласованному порядку. Изменения правил, области и ориентиров сохраняются с основанием и версией; выполнение не отмечается только по обещанию его сделать.

5. Вести журналы и развивать поддержку

На PoC настройте ежедневный сбор из доступных систем. Используются журнал задач, журнал улучшений и журнал показателей. Они имеют разные назначения и не дублируют один общий журнал PoC.

ИИ-инженер проверяет навык и исправляет ошибки извлечения или интерпретации; РП проверяет оценки и существенные расхождения. Автозапуск раз в сутки или просьба агенту задаются планом мониторинга; пропуски и пересборка остаются видимыми. Порядок недельного разбора — в процессе управления эффективностью.

Фактических исполнителей улучшений и их команды указывают в журнале улучшений. Если работу выполняет ИИ-инженер внедрения, поясняют, почему её не выполнял ИИ-инженер продукта, и при необходимости связывают с дальнейшим освоением. Полная передача сопровождения не является обязательной: назначенные участники, доступность и ресурс должны соответствовать принятому способу работы.

Показатели и правила расчёта

Определения показателей, формулы, состав данных и ограничения выводов о результате внедрения. Организация мониторинга, согласование оценок и разбор результатов — в гайде РП; настройка и проверка автоматического сбора — в гайде ИИ-инженера.

Матрица показателей

Показатели различаются по двум измерениям: **уровень применения** — предприятие или команда; **назначение** — результат, процесс или баланс. Метрики результата показывают достигнутый прогресс относительно согласованных целей внедрения. Метрики процесса и баланса нужны для управления: помогают обнаружить задержки, препятствия, ухудшение качества или рост нагрузки и затрат, чтобы выбрать действие.

Уровень	Результат — какой прогресс достигнут?	Процесс — где требуется изменить работу?	Баланс — какое ухудшение требует вмешательства?
Предприятие	Доли команд по этапам, сроки и трудозатраты по сопоставимым продуктам	Дни подготовки, часы адаптации и число препятствий	Качество, нагрузка служб и совокупные затраты
Команда	Срок поставки и трудозатраты на производство ПО	Автономность, идеальные часы объёма, медианы этапов и число доработок	Возвраты, пропущенные дефекты и нагрузка проверяющих

Уровень, область, назначение и границы выбранных показателей фиксируются в плане мониторинга: шаблон для продукта. Несопоставимые результаты разных продуктов сохраняются отдельными группами.

Основные понятия

Виды показателей

- **Результат** — достигнутый прогресс относительно целей внедрения.
- **Процесс** — показатели для управления работой: помогают найти задержки и препятствия, определить, что изменить.
- **Баланс** — показатели для управления качеством, нагрузкой и затратами: помогают обнаружить нежелательное ухудшение и выбрать вмешательство.

Назначение зависит от цели: дефекты — показатель баланса при ускорении поставки и результата при улучшении качества. Объём агентских задач и сокращение HITL не доказывают эффект сами по себе. Классификация адаптирована из Model for Improvement; выбор метрик и порядка проверки — в плане мониторинга.

Когорты и сроки наблюдения

- **Недельная когорта** — задачи, впервые переданные агенту с проработанными и согласованными требованиями в течение одной календарной недели.

- **Неделя формирования когорты** — календарная неделя, в течение которой в неё включаются переданные агенту задачи.
- **Период наблюдения за когортой** — заранее установленный срок наблюдения за этими задачами после недели передачи. Рекомендуем две следующие недели.
- **Окно оценки когорты** — весь интервал: неделя формирования когорты и последующий период наблюдения за ней. Рекомендуемое окно — три календарные недели.
- **Отсечка** — момент завершения окна оценки когорты. По событиям до неё фиксируется её итог.

Например: **неделя 25 — формирование когорты → недели 26–27 — период наблюдения за когортой → начало недели 28 — итоговый отчёт**. Окно оценки когорты охватывает недели 25–27.

Что фиксируется после отсечки

- До отсечки показатели **предварительные**, после — **итоговые**.
- Задачи без решения, принятой передачи человеку или отмены к отсечке относятся к **«Проблемным»**.
- Позднее завершение проблемной задачи не возвращает её в показатели нормального решения этой когорты.

План мониторинга продукта — шаблон.

Границы измерения эффекта

Целевой показатель относится к трудозатратам на производство ПО: техническому проектированию, коду, проверкам, тестированию, исправлениям и подготовке выпуска на основе согласованных требований. Бизнес-анализ и эксплуатация в него не входят. Для каждого потока фиксируется учитываемый состав работ; исходная оценка без агентов и оценка участия людей с агентами используют одинаковые границы.

Подготовка бизнес-требований до передачи задачи, эксплуатационные работы и общие затраты внедрения и сопровождения отражаются отдельно. Уточнения технического решения, проверки и исправления в ходе производства учитываются в соответствующей задаче. Состав работ определяется содержанием, а не должностью участника: техническая приёмка аналитиком может входить в производство, формирование бизнес-требований — нет. Если по исходным данным эти виды работ нельзя разделить, оценка помечается как несопоставимая и не включается в вывод о снижении трудозатрат на производство ПО.

Типовые метрики команды

Метрики выбираются под согласованные цели внедрения. Метрики результата показывают достигнутый прогресс относительно согласованных целей внедрения. Метрики процесса и баланса нужны для управления: помогают обнаружить задержки, препятствия, ухудшение качества или рост нагрузки и затрат, чтобы выбрать действие. Для выбранного показателя задаются поток, период наблюдения, исходная база, целевое значение или ориентир и допустимые отклонения. Связь целей с метриками и выбор показателей — в пояснении к плану мониторинга продукта; организация разборов — в гайде процесса.

Результат команды

Метрика	Значение и единица	Расчёт
Медианное время поставки (TTM)	Медиана календарных часов или дней от передачи задачи до выбранной точки поставки	P2 в сопоставимых условиях
Оценка экономии трудозатрат	Разность идеальных часов без агентов и оценённых часов людей на тот же выполненный объём; часы и доля в процентах показываются отдельно	P3 с учётом затрат всего потока

Процесс команды

Метрика	Значение и единица	Расчёт
Автономность по трудозатратам	Процент по формуле `(1 − часы людей / идеальные часы выполненного объёма) × 100%`, включая HOTL и доделки	P4
Входящий объём задач в идеальных часах	Сумма идеальных оценок всех N задач когорты	P1: V_n
Решённый агентами объём в идеальных часах	Сумма идеальных оценок задач с исходом S в окне наблюдения	P1: V_s, исходы
Медианная длительность этапа	Медиана интервалов между выбранными событиями начала и конца; календарное или рабочее время и единица заданы для этапа	Медианы этапов, ожидание ответа человека
Медианное время очереди HITL	Календарные и рабочие часы от передачи готовых материалов до начала содержательной проверки, отдельно по виду и адресату	Ожидания в процессе задачи
Медианное время внутренних ожиданий агентов	Календарные часы завершённых интервалов отдельно по причине: очередь, зависимость, лимит, среда или отказ	Ожидания в процессе задачи
Число циклов доработки задачи	Целое число завершённых циклов исправления по задаче	Определение и правила

Баланс команды

Метрика	Значение и единица	Расчёт
Доля возвратов с тестирования	Процент задач с хотя бы одним подтверждённым возвратом среди задач с завершённым первым проходом тестирования и полным окном наблюдения	B1

Метрика	Значение и единица	Расчёт
Доля задач с дефектами после приёмки	Процент принятых задач с подтверждённым дефектом, найденным после приёмки, среди задач с полным окном наблюдения	B2
Часы работы проверяющих	Сумма активных человеко-часов проверки за период, отдельно по роли; фактические или оценённые часы обозначаются явно	B3
Количество ожидающих проверок	Число открытых ожиданий проверки на дату среза, отдельно по роли	B3

Баланс измеряется до и после изменения на сопоставимых задачах при одинаковом способе выявления проблем. Допустимое отклонение и условия вмешательства задаются заранее. Отсутствие найденных дефектов при сокращении проверок не доказывает сохранение качества.

B1. Возвраты с тестирования

Доля возвратов = задачи хотя бы с одним подтверждённым возвратом на исправление / задачи с завершённым первым проходом согласованного тестирования и полным окном наблюдения × 100%. Числитель — подмножество знаменателя; при нулевом знаменателе — «н/д». Повторные возвраты одной задачи не увеличивают долю, но число циклов показывается отдельно. Возвратом считается подтверждённый дефект исходного согласованного объёма; новые требования, сбой стенда и ожидание данных выделяются отдельно.

По тем же задачам учитываются возвраты, обнаруженные в согласованном окне наблюдения; незавершённые проверки, ещё не созревшие задачи и известные по ним возвраты показываются рядом. Сравниваются типы задач, режим разработки, состав тестирования и критичность дефектов. Если тестирование стало тщательнее, рост возвратов нельзя целиком приписать переходу на HOTL.

B2. Дефекты после приёмки

Доля задач с пропущенным дефектом = принятые задачи с хотя бы одним подтверждённым дефектом, найденным после приёмки / принятые задачи с полным окном наблюдения × 100%. Числитель берётся из того же знаменателя; при нулевом знаменателе — «н/д». Окно после приёмки задаётся одинаково для сравниваемых задач. Для ещё не созревших задач показываются возраст и уже известные дефекты, без включения в итоговую долю.

Рядом показываются количество и критичность дефектов, место обнаружения — следующий этап, UAT или production — и полнота связи с исходной задачей. Один дефект, затронувший несколько задач, не размножается в числе дефектов; правило отнесения фиксируется заранее. Поздняя находка сохраняется в истории качества и отдельном наблюдении, не меняя исход S/H/C/P когорты задним числом.

B3. Нагрузка проверяющих

По каждой проверяющей роли показываются оценённые активные часы проверки за период и на проверенную задачу: сумма часов проверки / число задач с известной оценкой проверки. Рядом — покрытие оценками, количество обработанных задач, число и рабочий возраст открытых ожиданий. Периодная нагрузка учитывает работу с задачами любых когорт и обозначается отдельно от когортного эффекта; одна активность не дублируется в общих затратах.

Количество ожидающих проверок — число открытых запросов на проверку по каждой роли на дату среза; напоминание по тому же запросу не создаёт новую проверку. Рядом показываются рабочий возраст ожиданий и количество обработанных проверок.

Включаются ревью, разбор доказательств и последующие HOTL-проверки и замечания. Длительность ожидания не подменяет активное время человека. Источники и ограничения автоматической оценки — по правилам H_i. Снижение времени на задачу при перегрузке роли и растущей очереди не подтверждает допустимую общую нагрузку.

Объём, сроки и трудозатраты

P1 описывает размер потока, P2 и P3 — эффект в выбранных границах. Обозначения N, S/H/C/P и правила исходов приведены ниже.

P1. Объём задач когорты в идеальных часах

Объём когорты — суммарная оценка того, сколько активных человеко-часов потребовалось бы команде для выполнения переданного агенту объёма обычным способом, без агентов и без ожиданий. Это размер входящего объёма задач, а не время работы агентов, расход токенов или длительность календарной недели.

Для задачи i обозначим эту оценку E_i. Это оценка того, сколько идеальных часов потребовалось бы людям для выполнения согласованного состава работ без агентов. Основания — требования, контекст кодовой базы, сложность изменений и сопоставимые задачи.

В проекте выбирается один из двух способов оценки:

- **Агентом:** агент автоматически готовит оценку и её обоснование.
- **Командой:** участники команды оценивают задачу в принятом у них порядке; можно использовать существующую оценку или подготовить её для мониторинга.

При любом способе оценка проходит согласование с РП по плану мониторинга. Уточнение оценки участниками команды или РП входит в этот порядок и не образует отдельного способа оценки.

Для каждой версии сохраняются источник и автор оценки, дата, версия требований, учитываемый состав работ и краткое обоснование. Готовая оценка команды используется при совпадении единиц и состава работ. Story points не переводятся в часы без отдельно проверенного правила проекта. Для сравнения задач потока используется единый выбранный способ оценки; его изменение указывается в версии правил, а не выбирается по величине экономии.

Предварительная оценка может быть зафиксирована до передачи задачи в работу. Она помогает оценивать входящий объём, но не является условием запуска. **Итоговая оценка** фиксируется после приёмки результата, когда понятны согласованный выполненный объём и выявленная сложность работы. Если предварительная оценка была, её проверяют и при необходимости корректируют; если её не было, оценку впервые формируют после приёмки. При неизменном значении также отмечается переход к итоговой оценке.

При корректировке сохраняются исходное и новое значения, дата, автор и причина: например, обнаруженная сложность интеграции, уточнение предположений или изменение согласованного объёма. Изменение объёма отделяется от уточнения сложности прежнего объёма. Новые возможности не увеличивают задним числом объём, переданный на старте: исходные границы и добавленный объём показываются отдельно. E_i и H_i для расчёта экономии всегда относятся к одному составу работ.

Например, до начала задачи команда оценила работу в 8 идеальных часов. При выполнении выяснилось, что согласованное поведение требует обработки дополнительного состояния интеграции; после приёмки оценка обычной человеческой работы уточнена до 12 часов. Сохраняются обе оценки и объяснение разницы в 4 часа. Число попыток агента или длительность его работы сами по себе такую поправку не обосновывают: оценивается необходимая работа людей без агентов, без переноса на неё лишних действий и ошибок агента.

В предварительных сводках используется последняя доступная версия E_i. После приёмки — итоговая; её отсутствие показывается явно, а экономия по такой задаче остаётся предварительной. По незавершённым, переданным и отменённым задачам используется последняя обоснованная оценка переданного объёма с отметкой её статуса; она не заменяется оценкой только выполненной части. В сводке отдельно показываются число предварительных и итоговых оценок. Если оценки нет, действует правило неполных данных ниже.

На отсечке сохраняется версия оценки, использованная в итоговой сводке когорты. Последующие уточнения публикуются отдельной версией расчёта с основанием и прежними значениями; они не меняют фактические исходы задач и не превращают поздний результат в своевременное решение.

$V_n = \Sigma\ E_i$ по всем N задачам когорты. По завершении окна оценки когорты дополнительно показываются V_s, V_h, V_c, V_p — объёмы решённых, переданных, отменённых и проблемных задач; при полной оценке $V_n = V_s + V_h + V_c + V_p$. До отсечки вместо P используется U. Объём S показывает завершённый агентом объём, объём N — сколько команда доверила агентам.

Указываются идеальные человеко-часы и покрытие `число задач с оценкой / N`. Если оценено 8 из 10 задач, сумма восьми обозначается как **известный объём, покрытие 8/10**, а полный V_n — «н/д». Неоценённая задача не равна нулю. N и распределение исходов показываются даже при полном покрытии: рост объёма сам по себе не доказывает успешного завершения задач.

P2. Time to market

Для задачи $TTM_i = t_end_i - t0_i$. Начало — передача готовых требований агенту по правилу когорты. Конечная точка определяется проектом: например, влитие всех необходимых MR в интеграционную ветку, успешное завершение согласованного набора тестов на нужной ревизии или выпуск. Название показателя должно называть точку: «TTM до main», «TTM до успешного тестирования».

TTM считается в календарных часах или днях, включая ночи, выходные, очереди, HITL и доработки. По S показываются медиана, при необходимости среднее, число валидных наблюдений и покрытие относительно S. Для чётного числа наблюдений медиана равна среднему двух центральных значений отсортированного ряда. При отсутствии наблюдений — «н/д».

Конечная точка TTM и условия исхода S могут различаться: MR уже влит, но приёмка ещё не завершена. Тогда итоговый TTM успешного решения публикуется только после подтверждения S в окне оценки когорты. Промежуточные времена по остальным задачам относятся к диагностике, а не к успехам. H/C/P исключаются из TTM успешного решения, но их число, затраты и возраст незавершённой работы остаются рядом. Нельзя улучшать TTM, молча удаляя долгие задачи.

Если нужных MR несколько, конец интеграции — влитие последнего необходимого MR. Успешное тестирование — подтверждённый итог всех обязательных проверок целевой ревизии; результат теста другой ветки или старого коммита не подходит. Два разных конечных события могут отслеживаться отдельными показателями, без смешивания в одной медиане.

P3. Оценка сэкономленного времени

Для решённой агентом задачи $\Delta_i = E_i - H_i$, где H_i — оценённые активные трудозатраты людей на тот же состав работ, что включён в E_i. Для когорты $\Delta_s = \Sigma(E_i - H_i)$ по решённым задачам с обеими величинами. Отрицательная экономия сохраняется: она означает, что участие людей с агентом превысило оценку работы без агентов, а не ошибку формулы.

Здесь сравниваются только трудозатраты людей: идеальная оценка работы без агентов и оценка участия людей при решении задачи с агентом. Работа платформы, токены и стоимость платформы в E_i, H_i и разность Δ_i не включаются. Финансовую оценку с учётом стоимости платформы команда при необходимости настраивает отдельно при внедрении.

Доля экономии = Δ_s / Σ E_i × 100% при положительном знаменателе, по той же выборке. Проценты отдельных задач не усредняются. При неполных данных указываются сумма по известной выборке и её размер, а полная экономия когорты — «н/д». Отсутствие наблюдаемых действий не означает отсутствие работы человека.

Границы E_i и H_i фиксируются до Пилота. Если E_i оценивает работу после готовых требований, H_i включает весь соответствующий труд после передачи: уточнения, сопровождение, разработку человеком, ревью, тестирование, исправления и интеграцию. Подготовка до передачи показывается отдельно. Если команда включает подготовку в экономию, она включается в обе величины. Число часов подготовки не меняет t0 календарного TTM.

Нельзя считать экономию как V_n − часы всех задач: отменённый или незавершённый объём дал бы фиктивную выгоду. Рядом с Δ_s показываются часы H/C/P и общие затраты внедрения. При полном учёте можно добавить отдельный консервативный показатель: **оценка эффекта потока = $V_s - H_n$**, то есть ценность завершённого агентом объёма в идеальных часах минус оценённый труд по всем N задачам в тех же границах и до отсечки. Это отдельная оценка, а не экономия каждой задачи; ручное продолжение H и ограничения его наблюдения раскрываются.

Идеальные часы E и оценённые затраты H — две оценки с разными основаниями, а не измеренная реальность. Поэтому Δ обозначается **оценкой** высвобожденного времени, не денежной экономией или сокращением штата. На исторических человеческих задачах проверяют систематическое завышение/занижение оценок; изменение шкалы оценивания отмечается в сравнении когорт.

P4. Автономность по трудозатратам

A = (1 − H / E) × 100%, где H — фактические часы работы людей, E — идеальные часы выполненного объёма без агентов.

Для выбранной когорты в E включается только выполненный и принятый в окне наблюдения объём. В него могут входить задачи, завершённые агентом или доделанные людьми; факт ручного завершения должен быть подтверждён. Исход H «передано человеку» сохраняется даже после ручного завершения и не становится S. Незавершённый и отменённый объём в знаменатель не входит.

В числителе учитывается труд людей по **всем задачам этой когорты** в тех же границах производства ПО и окне наблюдения: уточнения, помощь агентам, HITL-ревью, асинхронные проверки HOTL, тестирование, исправления, ручное завершение, а также работа над отменёнными и незавершёнными задачами. Это сумма активных часов всех участников; параллельный час двух специалистов даёт два человеко-часа. Ожидание ответа или вычислений само по себе рабочим временем человека не считается. Общие расходы внедрения и сопровождения показываются отдельно по правилам границ.

Например, принятый объём оценён в 100 идеальных часов. Если люди потратили 30 часов, $A = 70\%$; если после ревью и HOTL пришлось добавить ещё 40 часов исправлений, $A = 30\%$. При 100 часах участия людей показатель равен 0%, при 120 — −20%. Отрицательные значения сохраняются; среднее процентов отдельных задач не используется.

Фактические часы берутся из доступного учёта, если он есть. При использовании автоматической оценки участия людей результат называется **оценкой автономности по трудозатратам**: источник и покрытие указываются явно, оценённые часы не выдаются за измеренный факт. Разные способы учёта не смешиваются молча. Нового обязательного табеля или запросов участникам о потраченном времени показатель не вводит.

При $E = 0$ показатель — «н/д». Если нет необходимых оценок объёма или данных о человеческом труде, полный показатель также не рассчитывается; пропуск не заменяется нулём. Пока запланированные HOTL-проверки и последующие исправления не завершены или не наблюдаются, показатель предварительный либо недоступен. Позднее выявленный человеческий труд отражается с периодом и версией расчёта; его нельзя скрывать за успешным завершением задачи. Исходы и отсечка когорты не переписываются ради поздних результатов.

Рядом показываются выполненный объём, фактические или оценённые часы людей, полнота данных, исходы задач и режимы контроля. Рост показателя отражает уменьшение участия людей; решение о смене HITL/HOTL принимается по действующему гейту.

Оценённые трудозатраты людей

Активное участие людей оценивается автоматически по артефактам задачи: сообщениям, изменениям требований и кода, решениям ревью и результатам тестирования. Источники, настройка и проверка оценивания описаны в гайде ИИ-инженера, разбор аномалий — в порядке согласования оценок.

H_i — сумма оценок человеческих действий по задаче в согласованных границах и окне оценки когорты. Это оценка, а не измерение продолжительности работы. Время агентов, токены и стоимость платформы в человеко-часы не переводятся.

По каким признакам оценивать

Сначала агент находит свидетельство выполненного человеком действия, затем оценивает его объём и сложность. Наличие большого MR само по себе не доказывает, что его прочитали. Учитываются авторство и содержание: текст, созданный агентом, не становится человеческой работой из-за публикации от имени участника.

Действие человека	Основание и признаки для оценки
Первичное ревью MR	Вердикт или содержательные комментарии; размер содержательного diff, сложность логики, число затронутых компонентов. Генерируемые файлы и форматирование учитываются отдельно от логических изменений
Повторное ревью	Проверка исправлений и новых изменений после предыдущего рассмотрения; полный MR повторно не начисляется автоматически
Обсуждение в чате или трекере	Объём и информационная насыщенность человеческих сообщений, вопросы, решения и необходимый контекст. Связанные сообщения объединяются в обсуждение; время между ответами не считается трудозатратами

Действие человека	Основание и признаки для оценки
Техническая проработка согласованных требований и спецификаций	Технические уточнения и ревью в пределах производства ПО; формирование бизнес-требований исключается. Число сценариев и развилок, сложность технического решения; отделяется создание человеком от проверки текста агента
Тестирование и приёмка	Отчёт или вердикт человека, число и сложность проверенных сценариев, найденные дефекты; собственное тестирование и просмотр доказательств платформы оцениваются раздельно
Ручные исправления и сопровождение	Изменения человека в коде, конфигурации и данных, диагностические действия с подтверждённым результатом; оцениваются по типу и сложности работы

Правила расчёта и полнота

Расчёт опирается на статусную модель задач, Git-процесс и маршрут работы с агентами: они определяют смысл переходов, решений и выбранный способ передачи задачи в работу. План мониторинга продукта задаёт метрики и необходимые подтверждения этих переходов; сопоставление событий с источниками выполняется по гайду сбора. Каталог событий сборщика ведётся в материалах мониторинга. Неописанный переход или отсутствующее подтверждение отмечаются как ограничение данных, а не достраиваются предположением.

Оценки используют зафиксированные правила выделения действий, признаки сложности, коэффициенты и диапазоны часов. Универсальная норма «минут на строку» не задаётся. Предварительная и итоговая E_i ведутся по правилам оценки объёма; H_i выводится из наблюдаемого участия людей и не рассчитывается как доля E_i или желаемая экономия. Уточнение E_i после приёмки не меняет независимо оценённую H_i.

Для каждого действия автоматически сохраняются ссылки на основания, исполнитель, связанные задачи, тип и объём работы, центральная оценка часов, диапазон и уверенность с объяснением. Эти записи нужны для воспроизводимости; вручную заполнять их не требуется. Оценки и извлечённые признаки сохраняются с версией правил, чтобы повторная агрегация не давала произвольных новых чисел.

- Одно действие учитывается один раз: обсуждение замечания в чате и MR связывается, повторные уведомления исключаются. Общая работа нескольких задач распределяется по правилу навыка без увеличения её суммы.
- H_i складывается из оценок уникальных действий; когортный итог — из оценок задач. Диапазон получается суммой нижних и верхних границ. Он отражает неопределённость модели оценки, а не статистический доверительный интервал.
- Для экономии показываются центральная оценка и диапазон: при $H_i \in [L_i; U_i]$ получаем $\Delta_i \in [E_i - U_i; E_i - L_i]$. Неопределённость самой E_i рассматривается отдельно. Если диапазон пересекает целевой порог, центрального значения недостаточно для уверенного вывода о достижении цели.
- Работа вне наблюдаемых систем, неоднозначное авторство и пропущенные источники отмечаются как ограничения. Отсутствие свидетельств не приравнивается к нулю. При неполном покрытии публикуются известная часть и размер выборки; полный итог — «н/д».
- Изменение коэффициентов версионируется. Для сравнения когорт используются одинаковые правила либо явно показывается изменение метода; пересчёт сохраняет предыдущую версию отчёта.

В недельной сводке показываются оценённые часы и диапазон по всем N задачам и отдельно по S/H/C/P, число задач с достаточными данными и ограничения. Для S добавляются среднее и медиана центральных оценок часов на задачу. Общие затраты настройки, обучения и развития процесса показываются отдельно.

По H оценки до передачи и ручного продолжения в окне оценки когорты показываются раздельно. Если ручное продолжение не наблюдается, это пробел данных. Поздняя работа может отражаться в календарной сводке, но не возвращает задачу в S. Календарный TTM и рабочие часы ожидания ответа рассчитываются по событиям отдельно от оценки активного труда.

Исходы, качество, ресурсы и этапы

Уровень команды: показатели её процесса выполнения задач и работы агентов.

Эти показатели помогают выбрать управляющее воздействие. Их рассматривают совместно: изменение одной величины само по себе не доказывает причину проблемы.

Порядок разбора и выбор действий.

Исходы задач

Обозначение	Итог в окне оценки когорты	Как понимать
N	Всего запущено	Неизменный знаменатель когорты
S	Решено агентом	Выполнены согласованные условия решения, включая выбранную конечную точку и обязательные проверки; обычные постановка, помощь, ревью и приёмка людей допустимы
H	Не решено агентом — передано человеку	Выполнение действительно принято человеком с контекстом и ответственностью; завершение ручной работы не требуется для фиксации H
C	Отменено	Есть явное решение и причина прекращения
P	Просрочено («Проблемные»)	К отсечке нет ни решения S, ни фактической передачи H, ни отмены C

Для итоговой когорты **N = S + H + C + P**. До отсечки вместо P учитывается U — ещё открытые задачи: N = S + H + C + U. В основном отчёте дополнительно показывается объединённое **C+P**, но составляющие сохраняются: отмена и затянувшийся процесс требуют разных действий.

Каждая задача имеет один исход. После принятой передачи выполнение относится к H, даже если позже люди завершили или отменили работу. Дальнейший результат не становится успехом агента S. Для повторного полноценного запуска того же объёма не создаётся новая запись ради улучшения статистики. Небольшая консультация или обычное ревью сами по себе не означают передачу; границу существенного ручного выполнения команда задаёт до Пилота.

P остаётся P независимо от последующего решения, передачи или отмены. Просроченные исключаются из показателей обычного успешного решения, но не из N и общих затрат. Это сигнал проблемы процесса, а не доказательство объективной сложности для агента: требования менялись, ответов не было, замечания повторялись, помощь пришла слишком поздно.

При сложности для агента решение о помощи или передаче принимается до истечения окна оценки когорты. Отметка «передать» без принявшего человека не спасает задачу от P.

Доли исходов = соответствующее число / N. При N = 0 доли не определены. Новые требования и существенное изменение объёма фиксируются отдельно; незавершённый прежний объём нельзя скрыть в новой задаче.

Замечания: три независимых распределения

По **решённым задачам S** отдельно считаются замечания: 1. К коду. 2. К автотестам. 3. По результатам тестирования человеком.

В каждом канале: **0 / 1 / 2 и более**, дополнительно **не проверено или нет данных / неприменимо**. Сумма пяти категорий одного канала равна S. Каналы пересекаются: одна задача может иметь 0 замечаний к коду, 2 к тестам и 1 по человеческой проверке. Складывать эти распределения нельзя.

Единица — отдельный подтверждённый недостаток исходного результата, требующий исправления. Один комментарий с несколькими независимыми недостатками может содержать несколько замечаний. Повтор того же недостатка или ответ в его ветке не увеличивает счётчик; возврат после неудачного исправления отражается в циклах доработки. Уточнения без требуемого исправления, отозванные/необоснованные замечания и новый объём отделяются от замечаний к исходному результату.

В эти распределения входят замечания людей клиента и сопровождения при ревью предъявленного результата. Собственная отладка агентов и диагностические сообщения инструментов не считаются человеческими замечаниями автоматически. Если человек подтверждает найденный инструментом недостаток, он учитывается один раз в соответствующем канале.

Замечания и циклы доработки — разные показатели. В одном цикле могут быть несколько замечаний. Граница замечания и правила устранения дублей согласуются до Пилота и одинаково применяются ко всем когортам.

«Без замечаний» можно поставить только после фактической проверки соответствующим способом. Отсутствие ревью не означает идеальное качество. Неприменимость объясняется заранее: например, задача не затрагивала автотесты и правила потока не требовали их изменения. Если нужные тесты не написаны, это пробел или замечание.

Доли 0/1/2+ рассчитываются от **проверенных применимых решённых задач** своего канала. Число непроверенных и неприменимых показывается рядом. Указываются режим ревью и способ человеческого тестирования: собственные проверки, оценка доказательств или сочетание. При смене режима сравнение оговаривается.

Существенные дефекты показываются отдельно от количества: одно критическое замечание нельзя интерпретировать как лучший результат, чем два косметических. Наблюдение дефектов после интеграции ведётся в одинаковом согласованном окне качества; оно может продолжаться после отсечки когорты и отмечается отдельно.

Число циклов доработки задачи

Единица — один завершённый цикл «пакет замечаний → исправление → повторная проверка». Показатель — целое число таких циклов по задаче в окне наблюдения. Повторное сообщение о том же пакете и смена запуска агента счётчик не увеличивают; незавершённый цикл показывается отдельно. Для сравнения групп задач можно использовать медиану по исходным значениям с указанием числа наблюдений.

Критичность замечаний

TBD

Токены агента: вход, выход, кэш, день и ночь

Суммируются все вызовы по задачам когорты в окне оценки, включая повторные и неудачные попытки с фактическим usage. Общие эксперименты и настройка, не относимые к задачам, учитываются отдельно.

Для каждой модели/провайдера и версии тарифа показываются: - вход без кэша; - выход; - чтение из кэша; - запись в кэш, если выделена источником/тарифом; - разбиение каждого показателя на день и ночь; - при наличии тарифов — стоимость и валюта.

Схема источника должна быть известна. Если общий вход уже включает cached tokens, кэш не прибавляется к нему повторно; для разбивки выделяются непересекающиеся категории. Если источник не позволяет отличить запись/чтение кэша или общий вход, указываются исходные поля и ограничение, а не выдуманная детализация. Разные модели и тарифы не усредняются перед расчётом стоимости.

Границы дня/ночи и часовой пояс определяются правилами тарификации проекта. **Рабочий календарь людей для этого не используется.** Ночной дисконт применяется только по согласованному тарифу; до его появления показываются объёмы без обещания экономии.

Предпочтительно распределять расход по времени usage-событий/тарификации. Если доступна только итоговая запись длинного вызова, действует заранее выбранное правило отнесения, соответствующее биллингу; приближение явно отмечается. Неопределённые события остаются отдельной строкой. Отсутствие данных не равно нулю.

Стоимость = сумма объёмов непересекающихся категорий × соответствующий тариф с его единицей расчёта. Оценка выгоды ночного тарифа — разница со стоимостью тех же ночных объёмов по дневным ставкам той же модели; это не доказательство роста продуктивности.

Медианы этапов

Цель — по задержке сразу определить, к какой роли обращаться за управляющим действием. **У каждого этапа ровно один ответственный.** Для времени работы агента это **ИИ-инженер**; для человеческой работы — конкретная роль, которая выполняет действие. Организатор отчётности не подменяет ответственного за этап.

Отчёт строится по тем же недельным когортам. Для этапов по умолчанию используются **календарные часы в зоне ответственности одной роли**; рабочие часы реакции человека — отдельный показатель. Это не активные трудозатраты: учитываются очередь, ночь и ожидание действия этой роли. Время, когда дальнейшая работа передана другой роли, относится к её этапу.

Каталог этапов

Это стартовый набор, а не обязательная для всех проектов статусная модель. Проект выбирает нужные этапы и связывает их с реальными событиями, даже если в трекере нет соответствующего статуса.

ID и этап	Ответственный	Начало → окончание	Как подтвердить и что показывает
M1. Подготовка результата агента	ИИ-инженер	Допуск готовой задачи через чат или заданный статус трекера `t0` → первое предъявление полного результата на ревью; учитываются только интервалы работы на стороне платформы	Полный комплект MR и доказательств, сообщение о готовности и назначение проверяющих. Draft MR не завершает этап. Ожидание ответа человека или действий DevOps учитывается отдельно
M2. Ревью кода	Разработчик клиента	Готовая ревизия MR передана разработчику на ревью → законченный вердикт этого раунда	Approve или законченный набор замечаний. Время draft до передачи на ревью и исправления агента не входят. Повторное ревью — новый интервал M2
M2-T. Ревью автотестов	Тестировщик клиента (автотесты)	Готовая ревизия тестов передана на ревью → законченный вердикт этого раунда	Те же правила, что для M2, но отдельная выборка для ревью автотестов
M3. Исправление результата агента	ИИ-инженер	Замечания переданы агенту на исправление → исправленный результат предъявлен для проверки	Учитываются только интервалы на стороне платформы, включая её внутренние проверки. Ожидание уточнений и последующее ревью людьми не входят
M5. Проверка результата человеком	Тестировщик клиента (ручное тестирование)	Проверяемая ревизия, данные и необходимые условия готовы, задание передано тестировщику → отчёт «пройдено» или «есть дефекты»	Ссылка на ревизию и отчёт. Подготовка стенда до передачи тестировщику и последующее исправление дефектов не входят. Повторная проверка — новый интервал M5
M5-G. Проверка результата агента	ИИ-инженер	При готовых условиях проверка поручена агенту → предъявлены результат и доказательства	Отдельная проверка агентами, включая автоматические запуски тестов. Отчёт с дефектами завершает интервал; исправление относится к M3. Проверки, уже включённые в M1/M3, повторно сюда не относятся

ID и этап	Ответственный	Начало → окончание	Как подтвердить и что показывает
M7. Ответ на вопрос по требованиям	Аналитик клиента	Явный вопрос передан аналитику → получен содержательный ответ	Рабочие часы по календарю адресата; подробный расчёт — ниже. Запросы другим ролям образуют отдельные показатели с названным ответственным

Для M1 команда заранее определяет комплект полного результата. Если нужны несколько MR, учитывается готовность всего обязательного комплекта. При блокирующем обращении к человеку или DevOps интервал агента приостанавливается; после ответа или устранения блокера продолжается. Время ожидания записывается на роль адресата. Если агент продолжает другую работу параллельно, её подтверждённое время сохраняется — ожидание по одной ветке не останавливает всю задачу автоматически.

M2 и M2-T начинаются с передачи готовой ревизии назначенному проверяющему. Для каждого проверяющего сохраняются его роль и события. Вердикт «требуются изменения» завершает раунд; исправление относится к M3, следующее предъявление открывает новое ревью. Если ревью проводит ИИ-инженер на PoC, для него создаётся отдельная строка с ответственным «ИИ-инженер». В одну медиану разные роли не объединяются. Merge может заменить отдельный вердикт только при подтверждённом выполнении ревью тем же ответственным; ожидание последующего merge относится к отдельному этапу.

Для тестирования точка начала определяется готовностью результата и передачей на проверку, а не самим фактом merge. Приёмку аналитиком, проверку человеком, работу агента и подготовку среды измеряют отдельными этапами с конкретной ролью. Тестирование до merge сохраняет свой реальный порядок. Отчёт с дефектами завершает проверку; циклы исправления не продлевают её до успешного результата.

При передаче работы между ролями граница подтверждается событием и ожидаемым действием. Для блокера со средой ответственным указывается DevOps клиента. Если невозможно восстановить, у какой роли находилась работа, этот интервал отмечается как неполнота данных и не попадает в медиану конкретной роли. Его длительность и количество показываются отдельно, а не исчезают из отчёта.

Расчёт медианы этапа

1. Отобрать задачи когорты, где этап применим и начался до отсечки. Управленческая диагностика использует не только S: завершённые интервалы H/C/P тоже помогают увидеть проблему процесса.
2. Восстановить интервалы по подтверждённым событиям и роли ответственного. Разделить интервалы при передаче работы другой роли; учесть паузы и возобновления. Для повторяющегося этапа и нескольких MR получить длительность **на задачу**: сумму непересекающихся интервалов одного этапа с одной ролью. Параллельные интервалы этой выборки объединить, чтобы не считать один час дважды.
3. В основной медиане этапа одна задача даёт одно наблюдение. Для M2 объединяются интервалы ревью кода, включая повторные; для M2-T — только ревью автотестов; для M3 — только исправления агента во всех циклах. Ответственный каждой строки фиксируется в плане мониторинга. Диагностика «на MR» или «на цикл» допускается отдельно, с явным названием и знаменателем; её нельзя смешивать с медианой на задачу.

4. Посчитать медиану длительностей задач с завершённым этапом. При чётном числе — среднее двух центральных значений. Рядом показать число наблюдений, число применимых задач, открытых этапов, прерванных этапов и задач с неполными данными. Если у задачи остаётся открытый интервал этого этапа, она не считается завершённым наблюдением основной медианы.
5. Для открытых этапов показать накопленное время ответственной роли до момента среза, для итогового отчёта — до отсечки; хотя бы медиану и максимум. Время пауз с передачей другой роли не включается; во время такой паузы счётчик исходной роли не растёт. Отмена или передача не притворяется успешным завершением этапа: оборванный интервал показывается отдельно. Известные завершённые интервалы такой задачи можно анализировать отдельно с явной пометкой.

Повторное открытие этапа до отсечки обновляет его длительность и состояние; окно оценки когорты остаётся прежним. Если событий недостаточно, выводится «н/д» и причина. Нельзя закрывать интервал текущим временем, концом недели или сообщением агента без подтверждения факта.

Медианы этапов **не складываются в TTM**: разные задачи образуют разные выборки, этапы могут перекрываться, а медиана суммы не равна сумме медиан. Для точного разбора конкретной задачи строится её временная линия. Для накопленной медианы используются исходные длительности, а не среднее недельных медиан.

Среднее и медиана ожидания реакции человека

Единица наблюдения — **завершённый интервал ожидания содержательной реакции на запрос агента**, а не задача целиком.

- Начало: явный запрос доставлен назначенному адресату в согласованном канале и содержит понятное ожидаемое действие.
- Конец: получен содержательный ответ, решение или выполнено запрошенное действие, позволяющее продвинуть работу.
- Автоматическая квитанция, уведомление о прочтении и «посмотрю позже» ожидание не завершают.
- Напоминания об одном запросе не создают новые интервалы и не сбрасывают начало.
- Если возникает новый самостоятельный вопрос после ответа, это новый интервал.

Длительность — сумма пересечений интервала с **рабочим календарём ответственного адресата**: рабочие часы, часовой пояс, выходные и праздники, согласованные перерывы. Для очереди одной роли заранее задаётся её календарь. При передаче вопроса другой роли закрыть участок прежнего адресата как переданный и открыть участок нового; передача не считается содержательным ответом. Длительности переданных участков показываются отдельно по роли.

Пример: при графике пн–пт 09:00–18:00 без исключаемых перерывов запрос в пятницу 17:00 и ответ в понедельник 10:00 дают **2 рабочих часа**. Праздничный понедельник изменит расчёт по календарю.

Среднее = сумма рабочих длительностей / число завершённых интервалов. Медиана — середина отсортированного ряда (для чётного числа — среднее двух центральных). Это интервальная статистика: задача с несколькими запросами даёт несколько наблюдений. Она не измеряет суммарную задержку задачи; параллельные ожидания нельзя просто складывать в её TTM.

Сводка строится по запросам задач когорты, доставленным в окне оценки когорты и завершённым до отсечки. Незавершённые запросы не попадают в среднее/медиану завершённых; рядом обязательны их количество и рабочий возраст (медиана/максимум), чтобы длительное молчание не скрывалось. Запросы, снятые без содержательного ответа, показываются отдельно и не записываются как нулевая реакция. Ответ после отсечки не меняет итоговую метрику реакции когорты.

Среднее и медиана считаются отдельно по каждой ответственной роли: аналитик, владелец продукта, архитектор, разработчик, тестировщик ручного тестирования, тестировщик автотестов, DevOps. В строке указывается ровно одна роль; общая медиана реакции разных ролей не используется как показатель управления. Для каждой роли расчёт ведётся по исходным интервалам, а не усреднением недельных значений.

Ожидания в процессе задачи

Определения ниже используются в ЖЦ задач, правилах Git и плане мониторинга. Статус трекера обозначает этап; его длительность сама по себе не равна очереди, активной работе или трудозатратам.

Границы и единицы

Очередь HITL начинается, когда материалы нужной ревизии готовы и запрос доставлен назначенному человеку; заканчивается началом содержательной работы. Для тестирования нужна также готовая среда и данные. Рассмотрение после старта до вердикта — другой интервал; его календарная длительность не равна активным часам проверяющего. Ожидание содержательной реакции заканчивается ответом/решением и может включать рассмотрение: эти показатели нельзя складывать между собой. Мгновенная содержательная реакция без отдельной работы имеет совпадающие границы, если это подтверждено.

Внутреннее ожидание агента начинается с блокирующего условия конкретного действия и заканчивается его снятием: зависимость, очередь готовых задач, лимит параллельных запусков, готовность тестовой среды, очередь CI/инференса либо инфраструктурный отказ. Исполнение запроса LLM, pipeline или развёртывания не называется очередью; когда это подготовка необходимой среды, её длительность входит в соответствующий интервал готовности. Общий инцидент учитывается у задачи только за время, когда он действительно препятствовал её работе.

Для каждой причины вычисляются **медиана календарных часов завершённых интервалов**, число наблюдений и покрытие событиями. Для HITL дополнительно — рабочие часы по календарю адресата. Время начала и конца хранится с часовым поясом; события разных систем приводятся к общей шкале. При недостоверных часах источник и погрешность отмечаются. Среднее, перцентили и пороги добавляются по потребности проекта, а не вместо раскрытия выборки.

Рядом обязательны **число открытых ожиданий и их возраст на срезе**, отдельно по причине и адресату/ресурсу. Закрытые отменой, передачей или заменой ревизии показываются отдельно от ожиданий, закончившихся началом работы/снятием блокировки. Неизвестное начало или конец не заменяется нулём; время до первого комментария допускается как отдельно названный показатель, но не выдается за точную очередь ревью.

Связи и перекрытия

Единица наблюдения — интервал конкретного запроса, запуска или блокировки с устойчивым ID, связанный с задачей, попыткой и при необходимости MR/SHA. Повторное уведомление и повторный суточный сбор не создают новый интервал. Новое предъявление после исправления — новый раунд, его история сохраняется. Время ожидания вне окна отчёта не теряется: открытый возраст считается от действительного начала.

Причины сохраняются отдельно, даже если действуют одновременно. Для итоговой длительности задержки задачи берётся объединение подтверждённых блокирующих интервалов внутри выбранной границы задачи; пересечения считаются один раз. Это время наличия блокировки, а не доказанная экономия от её устранения. Для аддитивной разбивки каждый отрезок получает одну подтверждённую основную причину по правилу проекта; при невозможности выбора — «несколько причин», без произвольного приписывания одной роли. Детальные распределения ожиданий разных запросов могут пересекаться и не складываются в TTM.

Готовность задачи к исполнению, очередь планировщика и очередь слота платформы — вложенные понятия: верхняя длительность не прибавляется к своим частям. Работа, продолжавшаяся параллельно ожиданию, сохраняется как работа; одна блокировка не означает простой всех агентов. Порядок очереди, недостающий слот и отказ инфраструктуры различаются по подтверждённому препятствию. Если причина меняется, интервал разбивается с сохранением связи.

Сбор выполняется ежедневно за прошедший день с возможностью пересборки, разбор — еженедельно по причинам, в общем цикле управления эффективностью. Полнота проверяется на PoC; в Пилоте существенные пробелы видны в сводках и ограничивают выводы.

Недельные когорты Пилота

Когорта — все задачи выбранного потока, **впервые переданные агенту в работу с проработанными и согласованными требованиями в течение одной календарной недели**. Момент передачи `t0` — первое подтверждённое разрешение взять готовую задачу по маршруту агента: сообщение уполномоченного человека в чате либо переход в статус трекера при выполнении заданных условий выборки. В обоих случаях определены задача и согласованный объём/версия требований. Учитывается время этого разрешения, а не начало вычислений; ожидание запуска входит во время решения. Если разрешение продублировано вторым способом, сохраняются один t0 и одна задача когорты.

Подготовка требований, обсуждение идеи и черновой вызов агента для уточнения постановки не включают задачу в когорту реализации. Произвольный статус «В работе» или назначение агента не заменяют выбранный сигнал допуска. Для маршрута через трекер подтверждением служит именно переход в настроенный статус с нужными условиями, а для чатового — поручение в установленном канале. Если событие передачи или согласованная версия требований не найдены, агент показывает нераспределённый запуск как пробел данных; инженер уточняет факт по источникам, не назначает удобную неделю. Количество таких запусков обязательно показывается рядом со сводками.

Перенос, повторный запуск агента, исправление, повторно открытая задача или смена исполнителя не меняют `t0` и когорту. Существенный новый объём оформляется отдельным решением и новой задачей; прежний объём сохраняет исход и затраты. Подготовка требований до `t0` при необходимости измеряется отдельным этапом, привязанным к будущей когорте задачи, и не

называется временем решения после передачи агенту. Поток разработки самих требований можно измерять отдельно, с собственным готовым входным заданием; смешивать его с реализацией нельзя.

До Пилота фиксируются календарь недель, год и даты когорты, часовой пояс, критерий готовности входных требований, критерий решения и конечная точка TTM. Все показатели задач относятся к неделе их передачи агенту, даже если ревью, тестирование или расход токенов произошли позже. Рекомендованное окно оценки когорты — **неделя передачи плюс две полные следующие недели**. Это не 14 дней от передачи каждой задачи.

Когорта недели 25	Что происходит
Неделя 25	Запускаются задачи, начинается сбор показателей
Недели 26–27	Период наблюдения за когортой: сбор исходов, замечаний, затрат и реакции людей
Конец недели 27	Фиксируется итоговый состав когорты
Начало недели 28	Публикуется итоговая сводка когорты 25

Отсечка — точный момент начала недели 28 в часовом поясе проекта; учитываются события до него. Другое окно оценки когорты допускается по предварительной договорённости, не продлевается задним числом ради неудобных задач. Пока окно оценки когорты открыто, когорта предварительная; после отсечки — созревшая. Открытые когорты не смешиваются с итоговыми.

Когортная оценка эффекта применяется на Пилоте. На PoC проверяются возможности и качество; статистика эффекта не является обязательной.

Сопоставимость и итоговый вывод

До Пилота выбираются сопоставимые типы задач, исходная база, целевые значения, правила качества и достаточный объём данных. Историческая база с только успешными задачами не описывает доли передач, отмен и просрочек; это ограничение фиксируется либо собирается подходящая база.

Относительное снижение = (база − Пилот) / база × 100%, при базе > 0, одинаковых метрике, границах и способе отбора. Человеческие часы, календарный TTM, токены и реакция людей не сворачиваются в один показатель.

Для заданного снижения на `r%` при сопоставимом базовом значении `B > 0` целевое значение равно `B × (1 − r/100)`. Например, для медианы TTM база и целевое значение должны относиться к одной конечной точке и сопоставимым задачам. При отсутствии сопоставимой базы процент снижения не объявляется измеренным фактом; можно задать абсолютный ориентир или сначала собрать базу.

Для накопленного результата используют сопоставимые созревшие когорты. Числа и суммы складываются; доли считаются от суммарного знаменателя; средние взвешиваются по числу наблюдений. Медианы нельзя восстановить усреднением недельных медиан: нужен расчёт по исходным значениям в системе-источнике. Если доступны лишь сводки, накопленная медиана отмечается как недоступная. Это не требует новой ручной карточки каждой задачи.

Хорошее время S при высокой доле H/C/P не доказывает успех всего потока. Рядом всегда видны полный состав, качество, объём и полнота данных. Численные пороги и достаточность выборки согласуются для проекта заранее; новые критерии формальной самостоятельности не добавляются.

По каждой сводке указываются период, версия правил, дата среза, источник и пропуски данных. Поздно полученное событие, фактически случившееся до отсечки, допускает исправление с историей и основанием. Реально позднее решение или передача не является исправлением данных и не меняет P.

Перед выводом о достижении цели всегда показываются N и исходы, полнота оценок/часов/событий, число наблюдений медианы, открытые ожидания и существенные дефекты. Эти сведения сопровождают выбранные показатели и не могут быть отключены ради более красивого отчёта. Нельзя сравнивать скорость двух когорт, не проверив различия объёма, типов задач и режима проверки.

Пример итоговой когорты

В когорте 25 запущено 10 задач. К концу недели 27: S = 6, H = 2, C = 1, P = 1. Итого N = 10; решено 60%, передано 20%, отменено или просрочено 20% (10% + 10%).

По шести решённым задачам: - Код: 3 без замечаний, 2 с одним, 1 с двумя и более; сумма 6. - Автотесты: 2 без замечаний, 1 с одним, 1 с двумя и более, 2 неприменимы; сумма 6, знаменатель долей замечаний — 4. - Тестирование человеком: 4 без замечаний, 1 с одним, 1 с двумя и более; сумма 6.

В начале недели 28 публикуется эта сводка. Ручное завершение двух H на неделе 29 не превращает их в S; позднее решение P не меняет его категорию. Часы и расход всех десяти задач внутри окна оценки когорты остаются видны.

Допустим, оценки всех десяти задач известны: V_n = 160 идеальных часов, из них S — 96, H — 32, C — 16, P — 16. Оценённые трудозатраты людей в одинаковых границах: S — 36 часов, H — 20, C — 4, P — 8; всего 68. Это автоматически полученные оценки, а не табельные часы. Оценка экономии на S: `96 − 36 = 60` часов, или 62,5%. Консервативный эффект потока: `96 − 68 = 28` часов. Выражение `160 − 68` использовать нельзя: оно включает незавершённый агентом объём в выгоду.

Допустим, диапазон оценённых затрат S составляет 30–42 часа. При фиксированном E_s = 96 диапазон экономии — 54–66 часов, или 56,25–68,75%. Например, при завышении оценки ревью из-за включения генерируемых файлов поправка получает обоснование и новую версию расчёта.

TTM шести S равен 24, 36, 48, 48, 72 и 96 календарным часам: медиана — 48 часов, среднее — 54. Все числа в этом примере учебные; они не задают норму или целевые значения проекта.

Учёт затрат развития

Создание и отладка временных скриптов, поддерживаемых тестов и требований для конкретных задач входят в затраты этих задач. Общие инструменты, обучение и развитие инфраструктуры учитываются отдельно. Одна активность учитывается один раз. Рост числа тестов или текстов сам по себе не доказывает эффект.

Порядок рассмотрения итогов и подготовки решения — в разделе о завершении Пилота.

Показатели предприятия

Область измерения — программа внедрения и включённые в неё продукты, команды и потоки задач. Для показателя указываются период, состав проектов, база сравнения, определения и полнота источников. Источники — планы и сводки проектов внедрения, результаты общих служб и работы офиса; конкретные единицы и границы задаются в плане мониторинга программы.

Результат предприятия

Показатель	Значение и единица	Основа измерения
Медианное время поставки (TTM)	Календарные часы или дни до выбранной точки поставки, отдельно по сопоставимым потокам продуктов	Правила P2 и исходные наблюдения команд; медианы разных продуктов не усредняются
Трудозатраты на производство ПО	Человеко-часы на согласованный объём работ в сопоставимых условиях	Фактическое или оценённое участие людей обозначается явно; показываются объём, база, изменение и ограничения
Доля команд, перешедших к Пилоту	Число команд с подтверждённым GATE-02 / число команд в области программы × 100%	Зафиксированное решение о завершении PoC и переходе к Пилоту на дату среза
Доля команд, завершивших Пилот	Число команд с подтверждённым GATE-03 / число команд в области программы × 100%	Зафиксированное решение о завершении Пилота на ту же дату среза

Охват внедрением оценивается по двум событиям: подтверждённым GATE-02 и GATE-03. У обеих долей один состав команд области программы; команда, завершившая Пилот, входит в оба числителя, доли не складываются. Дополнительный критерий «регулярного применения» для этих показателей не вводится. По GATE-03 отдельно видно основание завершения: достигнутые цели или поддержанный план их достижения; факт завершения этапа не подменяет метрики эффекта.

Процесс предприятия

Показатель	Значение и единица	Основа измерения
Медианное время подготовки команды к Пилоту	Медиана календарных дней от начала Подготовки до подтверждения GATE-02	Планы и решения команд; рядом число допущенных команд, количество и возраст ещё не завершённых подготовок
Трудозатраты на адаптацию практики	Человеко-часы адаптации проверенной практики для следующей команды	Работы по настройке, изменениям навыков и инструментов и проверке применимости; границы заданы заранее, фактические и оценённые часы различаются
Количество общих блокирующих зависимостей	Число незакрытых препятствий, мешающих нескольким проектам, на дату среза	Одно общее препятствие считается один раз; перечень затронутых команд и возраст показываются рядом

Показатель	Значение и единица	Основа измерения
Доля проектов с сопоставимыми измерениями	Число проектов с достаточными данными для согласованного сравнения эффекта / число проектов, чей срок оценки наступил, × 100%	Проверяются база, определения, зрелость когорт и полнота источников; критерии достаточности заданы в плане мониторинга

Для каждой доли публикуются числитель и знаменатель; при нулевом знаменателе — «н/д». Изменение области программы получает дату и основание. Отложенные, приостановленные и исключённые проекты показываются отдельно; наступивший срок оценки не исчезает из знаменателя из-за неудобного результата. Открытые внедрения до срока оценки не смешиваются с созревшими, рядом видны их число и возраст.

Баланс предприятия

Показатель	Значение и единица	Основа измерения
Доля задач с дефектами после приёмки	Процент принятых задач с подтверждённым поздним дефектом при полном окне наблюдения	Правила B2, отдельные сопоставимые группы продуктов; существенные дефекты и полнота связи с задачами показываются рядом
Часы работы офиса и сопровождения	Сумма человеко-часов поддержки, обучения, адаптации и подключения команд за период	Фактические или оценённые часы, состав работ и доступный ресурс специалистов; одна работа не учитывается дважды
Количество необработанных обращений	Число открытых обращений в сопровождение на дату среза	Очередь поддержки, правила статусов; повторное сообщение по тому же обращению не увеличивает число, возраст виден отдельно
Доступность платформы и инференса	Время доступности / согласованное время работы сервиса за период × 100%, отдельно по каждому сервису	Проверка предусмотренных операций; окно обслуживания и правила учёта простоев задаются заранее
Медианное время ожидания инференса	Медиана времени от приёма запроса в очередь до начала обработки, в миллисекундах или секундах	События очереди с единым временем и идентификатором запроса; рядом число наблюдений, пропуски и открытые ожидания
Затраты программы	Денежные расходы за период в выбранной валюте, отдельно на подключение и постоянную работу	Офис, обучение, адаптация, сопровождение, эксплуатация и инференс; общие расходы учитываются один раз

Объединение результатов команд допускается при сопоставимых определениях, единицах и границах. Проценты улучшения разных продуктов не усредняются в общий эффект; медианы рассчитываются по исходным наблюдениям. Показатели затрат программы не меняют границы трудозатрат на производство ПО в показателях команд.

Подготовка сводки предприятия и работа с различиями проектов описаны в гайде офиса внедрения ИИ, этапы и оценка программы — в описании внедрения на предприятии. Ритм работы офиса и его полномочия описаны в материале об офисе.

Автоматический сбор сведений и расчёт метрик

Материал объясняет, как ИИ-инженер настраивает сбор метрик из рабочих систем, где сведения часто не имеют единого формата: задачи и комментарии трекера, Git/MR, чат, журналы агентов, CI и тестовых сред. Команда продолжает работать в своих инструментах. Агент связывает подтверждения, готовит оценки и данные расчёта; человек проверяет существенные расхождения. Новые ручные таймшиты и универсальная форма каждого события не требуются.

Входы: ЖЦ задач, правила Git, план мониторинга, правила расчёта, системы и доступы из карточки инфраструктуры. Результат — версионированный навык сбора и вычислений, подготовленные наборы данных, проверенная дневная сборка и сводка с источниками и ограничениями.

Исполнители и результат

ИИ-инженер обследует источники, настраивает извлечение, связи, оценки и расчёты. Специалист команды внедрения может выполнить первоначальную настройку и обучить специалиста продукта. Поддержка не передаётся автоматически в день старта Пилота: конкретные работы и ответственность определяются в проекте по освоению.

РП организует доступность сведений, проверяет оценочные значения и существенные аномалии. Предметные специалисты уточняют смысл спорных действий и результатов. Сбор и подготовку отчётности выполняет агент внедрения; это функция агента, а не требование создать отдельную систему мониторинга.

На Подготовке фиксируются действующие процессы и подключения. На PoC по мере выбора метрик настраивают источники и проверяют вычислимость, текущие и предварительные целевые значения; к Пилоту проверяют ежедневный сбор и пробную недельную сводку. На Пилоте улучшают процедуру, особенно оценивание объёма, сохраняя версии и сопоставимость. Точный порядок — в этапах формирования целей.

От процесса и плана к сведениям

Для каждой метрики инженер устанавливает:

1. Какие события, оценки и расходы нужны по её определению в плане мониторинга.
2. Что они означают в фактическом процессе PoC или Пилота: поручение, начало работы, готовность MR, обращение HITL, вердикт, возврат, проверка, приёмка или передача человеку.
3. Где есть подтверждения и как связать их с задачей, запросом, MR/SHA, запуском, средой и прогоном.
4. Что можно извлечь достоверно, что требует оценки, где источник неоднозначен или недоступен.
5. Какая версия правил, процесса и источников применяется к периоду.

Соответствие смысла метрики реальным событиям фиксируется в плане мониторинга и карточках процессов. Технические запросы, API/поля, правила распознавания текстов и нормализации находятся в навыке сборщика. Не требуется описывать всю производственную деятельность: достаточно выбранного потока и сведений для его показателей.

Источники и достоверность

Источник	Что можно получить	Что нельзя считать доказанным автоматически
Трекер	Постановка и версии, связи задач, исполнитель, переходы, комментарии, исходы	Время в статусе не равно активным часам; «Тестирование» может включать очередь и подготовку среды
Git и CI	SHA, MR, draft/ready, проверки, замечания, approvals, merge, jobs и деплой	Создание draft не начинает формальное ревью; первый комментарий не определяет точное начало работы
Чат	Поручение, вопросы, ответы, предъявление результатов, решения и уточнения	«Сделаю» не доказывает завершение, «увидел» не завершает содержательный запрос
Журналы агентов и инференса	Запуски, очереди, ошибки, usage и связанные ID	Один запуск не равен одной задаче; отсутствие usage не равно нулевому расходу
Тестовые среды и прогоны	Готовность, данные, ревизия, начало/конец проверки, результат, диагностика	Зелёный прогон старой версии не подтверждает новую; технический успех не заменяет приёмку

Структурированные факты берутся из API и журналов. Из свободного текста агент извлекает **кандидаты в события**, сохраняя сообщение, автора, время, ссылку и основание интерпретации. Затем проверяет их по контексту процесса и другим источникам. Например, фраза «готово к проверке» без доступной ревизии и необходимых материалов не доказывает готовность тестирования.

Противоречия и неоднозначности не разрешаются выбором более выгодного числа. Они остаются в списке аномалий с влиянием на метрику и направляются подходящему участнику на уточнение. Не нужно вручную подтверждать каждую извлечённую запись: инженер проверяет правило на примерах, РП — оценочные величины и выделенные расхождения. Неизвестное значение отмечается как неизвестное.

Что настроить в навыке

Доступ и связи. Указать каналы, проекты, репозитории, журналы и разрешённые операции чтения. Связать задачу, версию требований, сообщения, MR/SHA, запускаемое действие, образ/digest, среду и проверку. Один устойчивый ID предотвращает дубли при повторной доставке. Общие MR и затраты нескольких задач распределяются по записанному правилу.

Происхождение данных. Хранить исходные ссылки/ID и доступные снимки, время события и время получения, версию извлечения, найденное соответствие и ограничения. Исходные факты, интерпретации и оценочные величины различаются. Изменения и удаления в источнике сохраняются в истории, а не исчезают из расчёта без объяснения.

Оценивание. Идеальные часы оценивает агент или команда по принятому способу. Участие людей рассчитывается из доступных действий и артефактов с сохранением оснований; оценка не называется зарегистрированным фактом. Значения проверяет РП по порядку плана мониторинга. После уточнения объёма сохраняются прежняя и новая оценки, причина и согласование.

Ожидания. По ЖЦ задач и Git-процессу различать очередь HITL, рассмотрение и активный труд; очередь готовых задач, зависимости, лимит параллелизма, готовность сред, очереди CI/LLM и отказы. Подтверждать начало/конец и блокируемое действие. Открытые интервалы сохраняются с возрастом, пересечения не складываются в TTM; правила расчёта.

Вычисления. Суммы, длительности, доли и медианы вычисляются кодом или запросами по подготовленному набору. Текстовая модель извлекает смысл и помогает оценить трудоёмкость, но не угадывает итоговую статистику. Для воспроизведения сохраняются набор фактов, согласованные оценки и версия алгоритма. Повторное извлечение моделью может дать иную интерпретацию: такая пересборка сравнивается с прежней и получает новую версию; повтор вычислений по неизменному набору должен давать прежний результат.

Публикация. Настроить журнал задач, журнал показателей и журнал улучшений. Метрики и ориентиры остаются в плане мониторинга; значения и проверки — в журналах и наборах. Новая обязательная база данных или дополнительный реестр для команды не вводится: техническое хранение выбирается по доступным системам.

Ежедневный сбор и пересборка

Сбор выполняется раз в сутки за прошлый календарный день. Часовой пояс, время и способ запуска задаёт план мониторинга: автоматически по расписанию либо по просьбе агенту тем же навыком. Например: «Собери сведения за прошлый день», «Пересобери сведения за 1–3 ноября 2026 года». Человек задаёт период, а не собирает показатели вручную.

При каждом запуске агент:

1. Определяет период и версии процессов, плана, навыка и правил; проверяет доступность источников. Сутки задаются от 00:00 включительно до 00:00 следующих суток, не включая правую границу.
2. Получает события и изменения записей. Для открытых задач и ожиданий перечитывает необходимые связанные сведения, включая начало до границы суток. Позднее подтверждение учитывается по времени самого события, а не загрузки.
3. Связывает факты между системами, устраняет повторы, сохраняет подтверждения и нерешённые противоречия. Изменённая запись обновляется с историей, не увеличивая количество действий.
4. Пересчитывает затронутые периоды и когорты по правилам отсечки. Поздно загруженный факт до отсечки может уточнить итог с новой версией; событие, наступившее после отсечки, не улучшает старый итог когорты.
5. Сохраняет запись запуска: период, версии, покрытие источников и дат, набор расчёта, ошибки и результат. При частичном сбое публикует доступное с ограничениями, а не отмечает день полностью собранным.

Пропущенные дни и частичные ошибки видны до догрузки. Пересборка заменяет расчёт периода новой версией, не прибавляет ту же работу повторно. Недельная сводка использует те же сведения и выбранное окно оценки; период извлечения и окно оценки когорты — разные вещи.

Проверка на PoC и регулярный контроль

Проверить навык на доступных реальных, исторических или учебных записях: обычный проход задачи; несколько MR; замечание и повторное предъявление; проверка старой ревизии; передача человеку и отмена; отсутствие оценки; совмещение ролей; открытое HITL и ожидание ресурса; параллельные и вложенные ожидания; границы суток/недели; пропущенные дни, повтор, поздний факт и недоступный источник.

Сверять выборку с первоисточниками: нет потерь и дублей, начало не позже конца, ревизии соответствуют проверкам, исходы и объёмы сходятся по правилам метрик, неизвестное не стало нулём. Медианы проверяются по исходным наблюдениям; оценки и их изменения имеют основания и подтверждения. Для тестирования правил отсечки не нужно ждать реальную когорту несколько недель.

Результаты проб связываются с работами POC-23 и POC-24. Частично работающий сбор не объявляется полным. На Пилоте существенные ошибки исправляются и затронутые периоды пересчитываются с сохранением истории.

Поддержка и исправления

На недельном разборе по процессу управления эффективностью отличают проблему продукта от ошибки сбора, интерпретации или оценивания. ИИ-инженер проверяет исправление на прежнем проблемном примере и связанных сценариях, сохраняет версию и отражает выполненную работу в журнале улучшений. РП согласует уточнённые оценки, методику и ориентиры, если они изменились.

Ошибку расчёта нельзя выдавать за улучшение агентов, а уточнение исходной базы — за полученный эффект. Когда данные разных редакций несопоставимы, это прямо указывается в журнале показателей; старые записи не переписываются без истории.

Проектирование тестовой инфраструктуры

Тестовая инфраструктура должна позволять агенту независимо разрабатывать и проверять несколько задач параллельно, а людям — принимать результаты без вмешательства в работу агентов и друг друга.

На этапе Подготовка команда определяет состав сред, необходимые инструменты и план их готовности. Существующие средства продуктовой команды переиспользуются, если обеспечивают нужную изоляцию и управление. Пример для трёх принимающих показывает, как оформить решение.

Этот материал описывает среды и средства выполнения проверок. Требования, тесты и данные сохраняются в рабочих источниках продукта. Выявленные улучшения назначаются в трекере; выполненные изменения и их проверки отражаются в журнале улучшений.

Среды, сборка и управление описаны в разделах 1–3. Моки, эмуляторы и приёмники эффектов и управляемое выполнение раскрывают конкретные механизмы тестирования; сводная таблица помогает выбрать применимые.

1. Два назначения сред: DEV и UAT

В этой методологии DEV и UAT обозначают назначение среды. Названия стендов в продукте могут быть другими.

	DEV — среда работы агента	**UAT — среда приёмки**
Кто работает	Только агенты. Люди подключаются для диагностики и помощи, когда у агентов что-то не работает или не получается	Человек, принимающий результат; агент имеет доступ для разбора проблем и замечаний и может помогать с подготовкой
Для чего	Развёртывание изменений по задаче, разработка и запуск тестов, проверка интеграций, исправление ошибок	Проверка разработанного по критериям приёмки, разбор замечаний и демонстрация результата
Кто управляет изменениями	Агент в пределах выделенной среды и согласованного процесса	По процессв продукте: принимающий, команда, CI/CD или агент. Развёртывание силами агента — один из вариантов
Режим работы	Среду можно пересобирать, перезапускать и сбрасывать в рамках текущей задачи	Во время приёмки версия и состояние сохраняются до согласованного изменения
Как выделяется	Из пула под параллельно решаемую задачу или независимый прогон	Существующий стенд, pre-prod или специально созданная среда; персональная среда предпочтительна там, где это удобно
Завершение работы	После сохранения результатов среда очищается и возвращается в пул	Следующее использование — по правилам выбранной среды и процесса приёмки

Количество независимых DEV-сред выбирается по планируемому параллелизму; 2–4 среды могут служить начальным ориентиром. Нужны также подходящие среды для приёмки — UAT. Наличие UAT обязательно, но создавать новый отдельный пул не обязательно. Сначала проверяется, где команда уже принимает изменения: на персональных стендах, общих тестовых средах или pre-prod.

Вариант UAT	Как готовится результат
Существующая среда принимающего	Принимающий выбирает ревизию ветки, запускает сборку при необходимости и деплоит опубликованный артефакт из хранилища своими силами либо через процесс команды
Общая тестовая или pre-prod среда	Развёртывание и приёмка выполняются по установленному процессв продукте
Новые среды для приёмки	Если подходящих сред нет, их можно поднять аналогично DEV; развёртывание поручить агенту или выполнять силами команды

При создании новых UAT ориентируются на число одновременно принимающих; в идеале у каждого своя среда. Например, для трёх аналитиков, формулирующих и принимающих задачи, можно предусмотреть три UAT. Принимающим может быть и другой назначенный участник.

Количество DEV выбирается по планируемому параллелизму. Если среда занята, новая задача ожидает свободную либо получает дополнительно созданную среду; чужое выполнение не перезаписывается. Динамически создаваемые среды допустимы, если одновременно доступны выбранные проектом независимые места работы.

При общей UAT или pre-prod согласуются состав проверяемой версии, время приёмки и порядок изменений, чтобы параллельная работа не разрушала проверяемое состояние. Персональные среды не являются обязательным условием внедрения; ожидание общей среды фиксируется как ограничение процесса.

Обязательное условие: доступ агента к UAT

Независимо от того, кто развёртывает результат и кому принадлежит среда, у агента должен быть доступ к используемой UAT для понимания и разбора проблем и замечаний принимающего. Одной ссылки на ветку для этого недостаточно.

Нужно обеспечить возможность увидеть фактически развёрнутую версию, проверить описанное поведение и получить связанные журналы, ошибки, конфигурацию и необходимые для воспроизведения данные. На подготовке проверяют этот доступ на конкретном сценарии и определяют, как передавать агенту шаги и контекст замечания.

Доступ для диагностики не означает обязательного права на развёртывание, сброс данных или изменение среды. Такие операции выполняет назначенный участник, штатный процесс команды либо агент в согласованных пределах. Для общего pre-prod отдельно фиксируются доступные агенту действия и порядок подключения ответственных за остальные операции.

2. Что означает независимость среды

Среда — это приложение вместе с состоянием, зависимостями и средствами управления. Разные адреса приложения сами по себе не обеспечивают независимость. Для существующей UAT или pre-prod описывают фактическую организацию и допустимые операции; обязательного повторения архитектуры DEV и права агента на сброс UAT не требуется. Влияние общих компонентов на приёмку учитывается в правилах использования среды.

Часть среды	Что должно быть независимо
Версия приложения и конфигурация	Можно независимо деплоить нужный артефакт из хранилища и задать конфигурацию выбранной среды
База данных и миграции	Записи, схема и её изменения не затрагивают соседние среды; можно отдельно восстановить исходное состояние
Очереди, события и фоновые обработчики	Сообщения и обработчики одной среды не потребляются другой; задания не выполняются над чужими данными
Кэш, файлы и объектное хранилище	Разделены ключи и данные; очистка одной среды не очищает другую
Моки, эмуляторы и приёмники эффектов	Разделены сценарии, состояние, журнал команд/операций и настройки отказов
Время, случайность и конфигурация	Часы, таймеры, генераторы и профили прогона управляются отдельно; изменения не затрагивают соседнюю среду
Управление и наблюдение	Развёртывания, сбросы, журналы и результаты проверок однозначно относятся к среде, задаче и прогону

Компоненты могут быть общими физически, если обеспечена и проверена логическая изоляция. Для неё могут использоваться отдельные экземпляры, базы, схемы, пространства имён или другие средства проекта. Для каждого общего компонента нужно объяснить, как разделяются состояние и управляющие операции.

Ресурсов должно хватать на выбранное число одновременных прогонов: логически изолированные среды не должны регулярно блокировать друг друга из-за нехватки доступной мощности.

Сборка, хранилище артефактов и деплой

Для всех стендов действует единая схема: исходный код → сборка → хранилище артефактов → деплой на выбранный стенд. Единственный источник бинарей для деплоя — хранилище артефактов. Это правило одинаково для DEV, UAT и pre-prod, независимо от того, кто запускает операцию.

1. Ветка или задача определяет исходный код; перед сборкой фиксируется точная ревизия.
2. Процесс сборки создаёт бинарный артефакт — например, пакет или контейнерный образ — и публикует его в хранилище.
3. Артефакт получает неизменяемый идентификатор и связь с ревизией и запуском сборки. Повторная сборка публикуется как новый артефакт.
4. Деплой получает выбранный артефакт из хранилища по точному идентификатору и применяет конфигурацию целевого стенда.

5. В записи деплоя сохраняются стенд, идентификатор артефакта, ревизия кода, конфигурация и результат проверки готовности. Проверки и замечания привязываются к этому артефакту.

Один опубликованный артефакт может независимо деплоиться на несколько стендов. Для приёмки выбирается конкретный артефакт с известными результатами проверок. Если требуется объединить изменения нескольких задач, формируется соответствующая ревизия исходного кода, выполняется сборка и публикуется новый артефакт.

Данные и настройки сценариев подготавливаются отдельно по воспроизводимым наборам и инструкциям. Стенд получает свою конфигурацию при деплое; идентификатор бинарного артефакта остаётся однозначным.

Деплой ревизии задачи и служебные ветки

Для выбранного потока агент должен уметь независимо проверить конкретную ревизию задачи на выделенной среде, без предварительного включения задачи в общую интеграционную ветку. Схема «разработчик проверяет на своём компьютере, затем команда выбирает задачи для единственной развёртываемой ветки» не обеспечивает это условие для параллельной агентской работы.

Полноценность среды определяется сценариями: приложение, нужные данные, миграции, фоновые процессы и зависимости должны позволять выявить значимые для задачи проблемы. Изолированные управляемые моки допустимы. Локальный запуск части приложения не заменяет проверки взаимодействий, которые в нём отсутствуют. Unit- и другие проверки без стенда сохраняются в своём контуре; они не заменяют необходимые стендовые проверки.

На Подготовке фиксируются ограничения CI/CD и выбирается решение:

Возможности CI/CD	Решение
Принимает исходную ревизию и целевую среду	Зафиксировать SHA, собрать и опубликовать артефакт, выполнить деплой по ID на выбранную среду
Собирает feature-ветки, позволяет выбирать артефакт для деплоя	Использовать артефакт нужного SHA и независимый деплой на выделенную среду
Работает только с заданными именами веток	Ввести служебную деплойную ветку для каждой нужной среды; переназначать её на выбранный commit задачи для запуска CI/CD
Только одна общая ветка и один стенд, независимость не обеспечена	Включить адаптацию CI/CD и подготовку сред в план; до выполнения не отмечать параллельный поток как готовый

Деплойные ветки — допустимый способ адаптации существующего CI/CD. Не требуется вводить их там, где исходная схема уже решает задачу.

Работа со служебной деплойной веткой

1. Зарезервировать среду под задачу/прогон; для UAT сохранить правила приёмки и подтверждение разрешённого изменения.
2. Зафиксировать feature-ветку и точный исходный commit SHA. По правилам проекта пересоздать или переназначить служебную ветку этой среды **точно на этот commit**, не объединяя в ней задачи и не добавляя продуктовые изменения.
3. Запустить привязанный к ней CI/CD. Сборка использует зафиксированный SHA, а не текущее значение имени ветки при позднем старте job. Артефакт публикуется под неизменяемым ID.

4. Перед деплоем проверить, что запрос всё ещё владеет резервированием среды и выбран для её текущего развёртывания. Сериализовать изменения одной среды и удерживать блокировку на время деплоя; устаревший pipeline не должен получить право изменить среду после нового запроса.
5. Выполнить деплой из хранилища, подготовить состояние, проверить готовность и фактическую версию. Сохранить цепочку **задача → feature-ветка → исходный SHA → деплойная ветка и запуск pipeline → ID артефакта → среда и запись деплоя → результаты тестов**.
6. Перед следующей задачей сохранить доказательства, освободить/переназначить среду и подготовить её данные, моки и время. Сама смена ветки не очищает состояние БД и не отменяет уже начавшиеся задания.

Резервирование имеет идентификатор или иной проверяемый признак актуальности. Порядок отмены старых jobs, блокировки деплоя и проверки актуального запроса фиксируется в инструкции. Одна только отмена pipeline без проверки на стороне деплоя не подтверждает защиту от запоздалого запуска.

Feature-ветка и MR сохраняют историю разработки и ревью. Служебная ветка не используется как источник продуктового merge. Пересоздание/принудительное обновление разрешается отдельному исполнителю только для указанных служебных веток; продуктовые и защищённые ветки сохраняют свои ограничения.

Если настройка CI требует дополнительных файлов, предпочтительно вынести её в подключаемую конфигурацию или другой механизм, не изменяющий выбранный исходный commit. Если это невозможно и создаётся отдельный сборочный commit, фиксируются оба SHA и полный diff; дополнительные изменения не должны незаметно менять продукт, а соответствие проверяемому MR отдельно подтверждается.

Для приёмки используется выбранный артефакт из хранилища. Если существующий CI/CD заставляет выполнить новую сборку для UAT, она получает новый ID; проверки прежнего артефакта не считаются автоматически проверками нового.

Что зафиксировать и проверить

Карта «среда → служебная ветка → pipeline → полномочия» и выбранный способ развёртывания записываются в Git-карточке, состав сред и доступов — в карточке инфраструктуры. Пример инфраструктуры продукта — кабинет «Меридиан Сервис».

До запуска зависимой задачи проверяется путь выбранной ревизии до среды. К третьей фазе PoC проверяются параллельные деплои разных задач, смена задачи на среде, защита от устаревшего pipeline и воспроизводимость версии/данных. Отдельная машина инженера не служит подтверждением этих условий.

3. Управление средами со стороны сопровождения

В DEV действия агента должны выполняться программно — через API, CLI, задания CI/CD или существующие инструменты проекта. В UAT этот набор применяется к операциям, порученным агенту; самостоятельное развёртывание принимающим или штатным процессом клиента допустимо. Для любой UAT обязателен доступ агента для диагностики. Для каждого действия задаются целевая среда, входные параметры, проверяемый результат и способ получить ошибку. Операция не считается успешной только потому, что команда запуска принята.

Набор возможностей для DEV и для автоматизируемых операций UAT:

Действие	Ожидаемый результат
Выделить или зарезервировать среду	Известны назначение, текущая задача или принимающий и статус занятости
Собрать и опубликовать артефакт	По точной ревизии создан артефакт, доступный в хранилище под неизменяемым идентификатором
Выполнить деплой артефакта из хранилища	На выбранном стенде доступен нужный артефакт; зафиксированы его идентификатор, ревизия, конфигурация и проверка готовности
Подготовить тестовые или демонстрационные данные	Загружен указанный набор; известны его версия и способ повторного получения
Настроить сценарии интеграций	Моки подготовлены для нужного прогона в выбранной среде
Управлять временем и регулярными операциями	Выполнено заданное перемещение времени или срабатывание; доступен результат бизнес-операции
Выполнить проверки	Сохранены результаты и доказательства, связанные с фактически развёрнутой версией
Получить диагностику	Доступны журналы приложения, агентов, фоновых заданий и взаимодействий с моками
Очистить и восстановить среду	Сброшены данные, сценарии моков, таймеры и отложенные события; восстановлено известное исходное состояние
Освободить среду	Результаты сохранены, среда готова к следующей работе

В записи прогона сохраняются применимые параметры: артефакт и ревизия, среда, набор данных, конфигурация, версии моков и эмуляторов, сценарий эффектов, начальное время, seed и последовательность конкурентных событий. Секреты остаются в штатном хранилище; в записи используются ссылки или идентификаторы.

Конкретные команды описываются в инструкции проекта. Если действие требует человека, указываются ответственный и порядок обращения. Для DEV планируется автоматизация необходимых агентам операций; самостоятельное развёртывание человеком в UAT само по себе не является недостатком и не требует обязательной автоматизации.

Порядок работы в DEV

1. Агент получает свободную среду и связывает её с задачей и прогоном.
2. Выбирает опубликованный артефакт нужной ревизии либо запускает его сборку и публикацию. Выполняет деплой из хранилища в выделенную DEV, готовит данные, моки и начальное время.
3. Выполняет задачу, проверки и исправления; сохраняет доказательства.
4. При сбое, который агенты не могут устранить, подключается назначенный человек.
5. После завершения работы сохраняет результаты и освобождает среду по согласованному правилу.

DEV не используется людьми для обычной приёмки или демонстраций. Диагностическое вмешательство человека отмечается, особенно если оно изменило состояние и требует повторения проверок.

Смена задачи: БД и восстановление тестовых данных

Смена ветки или бинарного артефакта не возвращает БД в исходное состояние. Предыдущая задача могла изменить схему и данные так, что новая ревизия с ними несовместима. Перед запуском приложения выбирается и выполняется способ подготовки БД.

Ситуация	Правило
Новая ревизия текущей задачи	Состояние можно сохранить, если история применённых изменений совместима с целевой ревизией. Применить недостающие миграции и проверить результат
Другая задача или ветка	По умолчанию восстановить известную исходную БД и данные, затем применить изменения целевой ревизии. Сохранение БД допускается только с подтверждением совместимости истории миграций и данных
Истории миграций разошлись, версия неизвестна или предыдущее применение завершилось частично	Остановить использование среды; восстановить совместимое исходное состояние по инструкции. Не продолжать применение вслепую
Занятая UAT или общий pre-prod	Изменения выполняются по правилам этой среды с подтверждением ответственного за приёмку. Если сброс недопустим, выделить другую среду либо выбрать проверенный путь обновления с сохранением данных

Исходное состояние — **версионированный комплект схемы и данных**, совместимый с целевой ревизией: снимок БД с известной историей миграций либо создание БД с нуля и загрузка заданного набора. У комплекта указаны источник, версия схемы, версия данных и инструкция восстановления. Снимок с чужими изменениями ветки не становится подходящим только потому, что он успешно восстановился.

Порядок сброса и подготовки

1. Зафиксировать целевые задачу, SHA и ID артефакта, среду и принадлежащую ей БД. Проверить резервирование и права на сброс; блокировка должна охватывать подготовку БД и деплой. Устаревшие pipeline не могут менять ни приложение, ни БД.
2. Сохранить доказательства предыдущего прогона и нужное для разбора замечаний состояние. Для UAT получить зафиксированное разрешение на изменение или завершение приёмки.
3. Остановить приложение, фоновые обработчики, расписания и другие источники записи в эту БД; завершить активную работу по инструкции проекта. Изолировать или очистить относящиеся к старому прогону очереди, кэш, файлы, моки и таймеры. Соседние среды продолжают работу.
4. Пересоздать только целевую БД/схему либо восстановить выбранный снимок. Проверить версию схемы и историю применённых изменений. Исходная история должна соответствовать поддерживаемой точке начала для миграций целевой ревизии; при отсутствии подходящего снимка использовать создание с нуля.
5. Применить изменения БД целевой ревизии в установленном порядке, автоматическим средством либо временным ручным процессом. Проверить итоговую схему и журнал применения. Не подменять выполнение миграций правкой записей об их применении.

6. Восстановить тестовые данные по рецепту для этой версии: снимок уже содержит базовые данные; дополнительные наборы загружаются в явно указанном порядке без дублей. Данные для проверки преобразования загружаются **до** соответствующей миграции, сценарные данные для готовой схемы — после. Задать начальное время и настройки моков.
7. Выполнить деплой выбранного артефакта из хранилища и запуск в порядке, определённом инструкцией проекта. До готовности схемы и данных приложение не обслуживает запросы и не выполняет фоновые операции. Проверить версию, smoke-сценарии, контрольные данные и отсутствие влияния на другие среды.
8. Сохранить результат подготовки: среда/БД, задача, SHA, ID артефакта, исходный комплект, применённые изменения, итоговая версия схемы и данных, исполнитель, время и доказательства проверок. Только после успешной проверки среда доступна задаче или принимающему.

При сбое миграции или загрузки данных среда получает статус «не готова». Сохраняются ошибка и фактическое состояние; повторный запуск допустим только по предусмотренному для этого состояния правилу. Иначе выполняется восстановление и повтор всей процедуры. Обратные миграции не считаются универсальным способом смены ветки: они могут быть необратимыми или терять данные; использовать их можно только по отдельно проверенной инструкции.

Сброс должен воспроизводимо давать заданные схему и данные, в том числе при повторном выполнении. Это не требование повторно применять уже выполненную миграцию: средство миграций должно учитывать историю применения.

Проверка на чистой БД не заменяет проверку обновления существующей. Для изменений схемы и преобразований данных нужен отдельный прогон с поддерживаемой предыдущей версии и характерными данными: применение изменений, сохранность и правильность преобразования. Если база восстанавливается уже в конечной схеме, такой прогон миграцию не проверяет.

Пример подготовки БД — смена задач на одной среде. Инструкция конкретного продукта связывается с его ревизиями и средами.

Порядок работы в UAT

1. Команда определяет, где принимается задача или набор изменений и кто готовит результат.
2. Принимающий, другой участник, штатный процесс команды или агент выполняет деплой выбранного артефакта из хранилища. Если нужного артефакта ещё нет, сначала выполняются сборка и публикация. Фиксируются адрес, идентификатор артефакта, ревизия, данные, готовность и ограничения; проверяется доступ агента для диагностики.
3. Человек проводит приёмку. При замечании передаёт агенту среду, версию, шаги, ожидаемое и фактическое поведение.
4. Агент использует доступ к этой UAT, чтобы разобраться в замечании, проверить состояние и собрать диагностику. При необходимости запрашивает помощь DevOps или готовит исправление.
5. Новую версию, данные или сценарии подготавливают по согласованному процессу. Агент может делать это по запросу, если такие операции ему поручены. Изменения фиксируются вместе с необходимыми повторными проверками.
6. Результат приёмки и замечания связываются с фактически проверенной версией и состоянием среды.

Для каждой приёмки фиксируются артефакт из хранилища, конфигурация целевого стенда, набор данных и настройки сценариев. Состав изменений в артефакте должен быть известен принимающему и агенту.

4. Программно управляемые моки и эмуляторы

Внешние сервисы, оборудование и получатели побочных эффектов включаются в единый порядок управления тестовыми зависимостями. Для них используются моки, эмуляторы и тестовые приёмники операций: агент задаёт сценарий, выполняет проверку, читает фактические обращения и результаты, затем сбрасывает контекст.

Адрес, порт и режим подключения задаются тестовой конфигурацией. Агент должен направить приложение на выбранную тестовую зависимость и увидеть фактически выбранное подключение. Если адрес или обращение к устройству зашиты в коде и перенаправление невозможно, до зависимых проверок требуется доработка этого механизма.

Общие требования

У каждой DEV- и UAT-среды должен быть собственный независимый контекст тестовых зависимостей. Контекст включает исходное состояние, профиль и версию протокола, ответы, данные, сценарий вызовов и событий, журнал операций, задержки и отказы. Один физический сервис допустим при подтверждённой независимости контекстов; иначе нужны отдельные экземпляры. Настройка или сброс одной среды не меняют соседние.

Через API, CLI или сценарный файл агент должен уметь:

- выбрать профиль, начальное состояние и сценарий, задать параметры;
- получить штатный ответ, ошибку, недоступность, задержку или отсутствие ответа для проверки тайм-аута;
- задать последовательность, например «два отказа, затем успех», а по применимости — повтор или повреждённый пакет;
- подать, задержать или повторить входное событие, показание, нажатие либо обратный вызов;
- прочитать запросы, команды и сообщения: параметры, адресатов, содержимое, число и порядок обращений, а также итоговое состояние тестовой зависимости;
- проверить ожидаемые и запрещённые операции, дубли, частично выполненные действия и восстановление после ошибки;
- сбросить состояние и историю отдельно для нужной среды или прогона, сохранить сценарий и журнал для повтора.

Для каждого выбранного сценария известны команды подготовки, проверки и сброса. Глобальный переключатель ошибок на несколько сред и ручная перенастройка соседнего стенда этим требованиям не соответствуют.

Применение к разным зависимостям

Зависимость	Что подготавливается	Что проверяется
Внешний сервис	Мок с заданными данными, ответами, отказами и асинхронными событиями	Запросы, результат обработки ответов, тайм-ауты, повторные попытки и восстановление
Получатель побочного эффекта	Почтовый/SMS-приёмник с API чтения, платёжный sandbox или эмулятор, отдельные тестовые БД, каталоги/bucket и очереди	Адресат, содержимое и число сообщений, данные и фиксация/откат транзакций, файлы, списания и отсутствие лишних операций

Зависимость	Что подготавливается	Что проверяется
Оборудование	Эмулятор интерфейса и поведения сканера, терминала, контроллера, датчика, принтера или другого устройства с профилем и версией протокола	Входные события, команды приложения, порядок обмена, занятость, отключение и восстановление

Попытки и завершённые эффекты наблюдаются отдельно: запрос платежа — в журнале вызовов, итоговое списание — в состоянии sandbox. Успешный ответ функции сам по себе эффект не подтверждает. Повтор операции не должен незаметно создавать второе письмо или списание. Назначения и права тестового контура исключают реальную отправку, оплату или управление оборудованием.

Для автономной разработки и тестирования аппаратных сценариев нужен эмулятор. Агент не получает доступ к физическому устройству. При недоступности эмулятора прогон завершается ошибкой без переключения на реальное оборудование; его адреса, порты и доступы не включаются в контур агента. При отсутствии эмулятора его создание включается в план, а соответствующие сценарии временно исключаются из автономного потока. Физическое устройство отдельно проверяет человек.

Соответствие реальной зависимости

Мок или эмулятор подтверждает поведение приложения в заданных условиях. Совместимость с реальным внешним сервисом проверяется отдельно в доступном тестовом контуре или другим согласованным способом. Если контур общий и не допускает изоляцию, такие проверки выполняются по отдельному расписанию.

Профиль устройства сверяет назначенный специалист: по документации протокола, эталонным обменам и отдельным аппаратным проверкам. В результатах перечисляются воспроизводимые и непроверенные свойства. Электрические характеристики, реальные помехи, механические отказы и фактическая производительность не считаются подтверждёнными эмуляцией. Пример программно создаваемого профиля устройства — Android Virtual Device; для конкретного оборудования нужен свой применимый эмулятор.

5. Управляемое выполнение сложных сценариев

Для выбранных сценариев определяются конкретные средства управления. Ниже перечислены механизмы; применимые средства должны быть доступны агенту программно до зависимой задачи. Для каждого известны команда или API, входные параметры, признак завершения, результат и способ сброса. Полный набор не требуется продукту, в котором соответствующих сценариев нет.

Время и глобальный контекст приложения

Для прогона задаётся профиль контекста: начальное время, часовой пояс, локаль, feature flags, переменные окружения, настройки арендатора и пользователя. Агент должен уметь выбрать профиль, применить его до запуска приложения и получить фактические настройки без секретов в отчёте. В ходе сценария время продвигается через отдельный механизм управления.

Если настройка или singleton меняет весь процесс и не имеет надёжного сброса, тест запускается в отдельном процессе или контейнере. Разные глобальные профили не применяются параллельно в одном общем процессе. После прогона снимаются временные обработчики и подписки, очищаются

реестры; следующий запуск получает исходный профиль. Для пользователя и арендатора используются отдельные тестовые учётные данные и данные; проверяется отсутствие переноса контекста между запросами.

Виртуальное время и регулярные операции

Если в выбранных задачах есть операции по расписанию, таймеру или истечению срока, их нужно проверять без ожидания реальных часов, дней или месяцев.

На обследовании перечисляются такие сценарии: например, истечение срока заявки, регулярное начисление, повторная попытка после задержки, напоминание или переход процесса по таймеру.

Подход выбирается по архитектуре:

Механизм	Что обеспечивает	Что нужно проверить
Виртуальное время приложения	Задание текущего времени и его продвижение внутри отдельной среды	Бизнес-правила, фоновые обработчики и связанные компоненты используют согласованное время
Управление таймерами и заданиями	Программный запуск нужного таймера или операции без ожидания, например управление таймерами процесса в Camunda	Срабатывание запускает реальную бизнес-логику и даёт наблюдаемый результат
Другой механизм сжатия времени	Ускоренное прохождение периода или последовательности операций	Понятно, какие свойства сценария сохраняются, а какие этим способом не проверяются

Не обязательно применять один механизм ко всему приложению. Для каждого сценария достаточно подходящего инструмента, но его область действия и ограничения должны быть описаны.

Управление временем внутри профиля

- Изменения ограничены выбранной DEV- или UAT-средой.
- Агент может задать начальное состояние, выполнить продвижение или запуск, дождаться завершения обработки и получить доказательства.
- Согласованы часы приложения, планировщика, движка процессов и моков там, где их поведение связано. Часовой пояс также задан явно.
- Повторные запуски, пропущенные интервалы и отложенные события обрабатываются по понятному правилу; сброс среды очищает накопленное состояние.
- Сценарий можно повторить из исходных данных и настроек.

Смена системных часов общей машины не подходит для независимого управления средами. Ручное присвоение записи конечного статуса также не заменяет проверку таймера и бизнес-операции.

Прямой запуск задания может проверять бизнес-логику, но сам по себе не доказывает правильность расписания. Если важны момент запуска, границы срока или повторные срабатывания, для них предусматриваются отдельные проверки.

Для функций, которые только читают текущую дату, нужен задаваемый источник времени: фиксированный момент, часовой пояс и возможность перейти через границу суток, месяца, года или применимого календарного правила. Для тайм-аутов проверяются также часы измерения

длительности и таймеры; изменение календарной даты само по себе их не проверяет. В сценарии задаются начальный момент, продвижение времени, ожидаемые срабатывания и подтверждение завершения обработчиков.

Если проверка проходит через браузер и сервер, управление временем задаётся для обеих сторон по применимости. Например, Clock API Playwright управляет браузерными часами и таймерами, но сам по себе не меняет время сервера, БД или внешнего сервиса.

Случайность и недетерминированные данные

Для случайного выбора, генерации идентификаторов, перемешивания и вероятностных ветвлений нужен управляемый тестовый источник:

- задание seed и отдельного состояния генератора для прогона;
- возможность выдать конкретную последовательность значений, чтобы принудительно пройти нужную ветку или коллизию;
- сохранение seed, версии генератора и входных данных, а при нескольких источниках — их отдельных настроек;
- набор прогонов с разными seed и сохранение каждого неуспешного варианта для повторения.

Один seed не обеспечивает повтор при изменившемся порядке конкурентных вызовов: тогда нужны отдельные генераторы по участникам или запись фактически выданных значений и порядка их потребления. Детерминированная подмена применяется только в тестовом режиме и не подменяет криптографическую случайность рабочего контура. Пример отдельного генератора с собственным состоянием и seed — Python Random.

Асинхронность, очереди и многопоточность

Для сценариев с фоновой обработкой и гонками требуются:

- управляемая подача сообщений, остановка и продолжение обработчика в выбранной точке, барьеры или защёлки для согласованного старта конкурирующих действий;
- возможность задать задержку, повтор, перестановку или пропуск сообщения, разрыв соединения и перезапуск участника, если такие случаи допустимы для протокола;
- ожидание наблюдаемого условия завершения с ограниченным сроком вместо фиксированного `sleep`: статус задания, число обработанных сообщений, завершённая транзакция или другой проверяемый результат;
- журнал с идентификаторами задач, сообщений и обработчиков, позволяющий восстановить порядок существенных событий;
- сценарии одновременного изменения одних данных, повторной доставки, отмены и восстановления после прерывания по применимости.

Гонка воспроизводится заданным чередованием действий через барьеры, тестовые точки остановки или управляемый планировщик. Если стек поддерживает запись и повтор расписания, оно сохраняется с результатом; пример такого инструмента — Coyote. Дополнительные прогоны с реальными потоками проверяют поведение штатного выполнения. Многократный успешный запуск без управления порядком не заменяет воспроизведение известной гонки.

Удержанное состояние и последовательности действий

Для функций, зависящих от предыдущих вызовов, нужен программный способ создать известное исходное состояние и выполнить заданную последовательность команд:

- подготовка или восстановление версии набора данных, сессии, кэша, счётчиков и состояния фоновых обработчиков;
- чтение проверяемого состояния после каждого значимого перехода через API, диагностический снимок или запрос к тестовому хранилищу;
- сценарии «повторить команду», «вызвать в другом порядке», «прервать и продолжить», «перезапустить процесс и восстановиться»;
- подтверждение, что после очистки следующий прогон не наследует данные, незавершённые задания и подписки предыдущего.

В доказательства входят начальное состояние, последовательность действий и ожидаемые переходы. Для восстановления после сбоя проверяются отдельно сохраняемое состояние и исчезающее состояние процесса; чистый перезапуск не заменяет этот сценарий. Смена схемы и набора данных выполняется по порядку восстановления БД.

Сводка выбора механизмов

Особенность функции	Что должно быть доступно программно
Случайность	Seed, заданная последовательность значений, сохранённый неуспешный вход
Время и глобальный контекст	Профиль времени, часового пояса, локали и настроек; продвижение часов и таймеров; отдельный процесс при общих настройках
Побочные эффекты	Тестовый приёмник, чтение фактических операций, проверка дублей и очистка
Асинхронность и многопоточность	Подача событий, барьеры или точки остановки, условие завершения и трасса порядка
Внешние системы и оборудование	Независимые управляемые моки и эмуляторы с заданными сценариями и журналом
Удержанное состояние	Подготовка и чтение состояния, последовательность действий, сброс и восстановление

6. Если инструментов пока нет

Отсутствие средств управления — результат обследования и основание для плана доработок. По выбранным сценариям нужны работы на конкретные механизмы из разделов 4–5: моки и эмуляторы, приёмники эффектов, управление временем и случайностью, конкурентными событиями, состоянием и конфигурацией.

Это работы по обеспечению тестируемости, а не расширение Пилота на задачи реализации нефункциональных требований. Их трудозатраты учитываются отдельно как подготовка инфраструктуры.

Автоматические миграции: развитие без блокировки старта

Автоматические миграции — версионированные изменения схемы и данных, которые применяются штатной командой или процессом деплоя с учётом истории выполнения. Они связаны с ревизией приложения; для каждой БД сохраняется результат применения.

Как требования и автотесты, автоматические миграции не обязательны для начала внедрения, но существенно влияют на его результат. Без них каждое изменение БД требует ручной координации, обращения к специалистам и ожидания HITL. Это увеличивает трудозатраты и время решения задач, даже если код агент пишет быстро.

Если автоматических миграций нет или они покрывают не все компоненты:

- Зафиксировать состояние в карточке инфраструктуры и инструкциях продукта: какие БД покрыты, какие изменения выполняются вручную и где сохраняется их история.
- Для ближайших задач подготовить временный порядок: версионированный скрипт/инструкция, исходная и целевая схема, исполнитель, запрос HITL, подтверждение применения и восстановление при ошибке. Изменения «по памяти» не являются воспроизводимым процессом.
- Включить внедрение или дополнение миграций в план наряду с автотестами и требованиями: результат, ответственный, срок, зависимые сценарии и доказательство готовности. Работы могут идти параллельно PoC и Пилоту.
- Проверить автоматизацию на создании чистой БД, обновлении поддерживаемой предыдущей версии с данными, учёте уже выполненных изменений и обработке сбоя. Подключить инструкции к навыкам или заметкам агента.

В чек-листе проверяются временный рабочий порядок и план развития, а не обязательное наличие автоматизации. До задачи, меняющей БД, должен быть обеспечен применимый способ изменения, проверки и восстановления. Ручное участие в продуктовых задачах включается в трудозатраты людей, ожидание ответственного — в показатели ожидания реакции; создание общего инструмента учитывается отдельно как развитие инфраструктуры.

План готовности

Возможность и сценарии, которым она нужна	Есть сейчас / переиспользуем	Что создать или изменить	Ответственный	Срок и зависимый этап	Как подтвердим готовность	Временное ограничение
DEV-среды по выбранному параллелизму	…	…	…	…	Независимые параллельные деплои артефактов из хранилища	…
Сборка, хранилище и деплой	…	Публикация артефактов и деплой по неизменяемому идентификатору	…	До первой задачи PoC	Связь ревизии, артефакта, деплоя и проверок прослеживается	…
UAT: существующие или новые	…	При необходимости поднять по образцу DEV; определить способ развёртывания и доступ агента	…	До приёмки	Результат доступен человеку; агент может разобрать замечание в этой среде	…

Возможность и сценарии, которым она нужна	Есть сейчас / переиспользуем	Что создать или изменить	Ответственный	Срок и зависимый этап	Как подтвердим готовность	Временное ограничение
Контексты моков и управление ими	…	…	…	До проверок соответствующих интеграций	Разные сценарии одновременно в разных средах	…
Управление временем / таймерами	…	…	…	До задач с регулярными операциями	Период проходит без реального ожидания, результат подтверждён	…
Эмуляторы оборудования и приёмники эффектов	…	Профили устройств, события, отказы, журнал и сброс; тестовые назначения	…	До зависимых агентских задач	Агент воспроизводит сценарий без физического устройства и реальных эффектов	Без эмулятора — исключить эти сценарии из автономного потока
Случайность, конкуренция и состояние	…	Seed, барьеры/события, профиль конфигурации, подготовка и сброс	…	До соответствующих сценариев	Повтор известного сбоя и отсутствие состояния соседнего прогона	…
Данные, сброс и диагностика	…	…	…	…	Повторяемый прогон из известного состояния	…
Автоматические миграции БД	Автоматически / частично / вручную	Версионирование изменений, применение и учёт истории, обработка сбоев	…	Срок по плану; временный способ — до первого изменения БД	Чистая БД и обновление предыдущей версии проверены, результат применения записан	Скрипт, исполнитель и HITL до автоматизации

Необязательно завершать все работы до начала онбординга PoC. Однако до начала зависимых сценариев нужные инструменты должны быть готовы. К третьей фазе PoC инфраструктура должна обеспечивать проверку выбранных типов задач Пилота.

Если срок готовности не совпадает с планом, команда выбирает: изменить сроки, явно сузить область либо согласовать временный ручной способ с его стоимостью и ограничениями. Такой способ не предоставляет агенту доступ к физическому оборудованию: до появления эмулятора соответствующие проверки выполняет человек отдельно, а автономный поток ограничивается доступными сценариями. Непроверенный сценарий не отмечается как пройденный; при сохранении его в области остаётся план создания нужного инструмента.

7. Как подтвердить готовность инфраструктуры

Проверка проводится на выбранном для проекта числе сред и сохраняется со ссылками на результаты.

Проверка	Признак готовности
Источник и идентичность артефакта	Для деплоя в DEV, UAT или pre-prod указан артефакт хранилища; его идентификатор совпадает с фактически запущенным, а проверки привязаны к нему
Параллельные задачи в DEV	На разных стендах одновременно выполнен деплой артефактов разных задач из хранилища; данные и результаты не смешиваются
Сброс одной DEV	Вторая DEV и занятые UAT продолжают работу, их состояние не меняется
Смена задачи с несовместимой историей миграций	Восстановлен совместимый исходный комплект; применены изменения новой ревизии, загружены нужные данные и пройдены проверки. Следов предыдущего прогона нет
Обновление БД с данными	Проверено преобразование с поддерживаемой предыдущей версии; данные сохранены по правилам задачи. Указан фактический способ: автоматический или ручной
Независимые моки	Одна среда получает успех, другая — задержку или ошибку; настройки, журналы и сбросы изолированы
Приёмка в UAT	Выбранная среда, включая существующий pre-prod, готова для приёмки; версия и порядок изменений известны; работа в DEV не перезаписывает UAT
Доступ агента к UAT	Агент подключается к фактически используемой среде, проверяет поведение и получает диагностику по замечанию независимо от того, кто выполнял развёртывание
Операции UAT, порученные агенту	Если развёртывание или подготовка данных поручены агенту, он изменяет только целевую среду по согласованному запросу; результат виден в журнале
Регулярная операция	Агент вызывает нужное событие без реального ожидания и предъявляет результат; время и задания соседней среды не меняются
Оборудование и эффекты	События и отказы устройства заданы через эмулятор; проверены команды и побочные эффекты; недоступность эмулятора не включает реальное устройство
Случайность и конкурентность	Неуспешный вход/seed и известное чередование действий воспроизводятся; ожидание завершения не зависит от произвольного sleep
Удержанное состояние и глобальный контекст	Подготовка, последовательность действий, перезапуск и сброс проверены; следующий прогон не наследует состояние и настройки
Повторный прогон	После сброса тот же набор версии, данных и сценариев даёт воспроизводимый проверяемый результат

Неприменимость проверки обосновывается отсутствием соответствующих сценариев в выбранной области. Если сценарии есть, отсутствие инструмента оформляется работой в плане.

Итог Подготовки: перечень DEV/UAT, карта общих и изолированных компонентов, инструкции управления, подтверждения готовности и план незакрытых работ. Ответственность закрепляется за конкретными участниками проекта.

8. Инфраструктура запуска тестов

Инфраструктура должна позволять воспроизводимо запускать проверки и сохранять результаты. Виды тестов, старт без готового набора и порядок его развития описаны в стратегии тестирования.

Контур	Что предусмотреть
Unit и другие проверки без стенда	Тестовый раннер локально или в CI, воспроизводимый образ с инструментами, точная ревизия и отчёты. Для проверок взаимодействия с БД — отдельная тестовая БД или временный контейнер с изоляцией прогонов; такие тесты относятся к интеграционным
API и браузерные UI-проверки приложения	Выделенная DEV, деплой артефакта из хранилища, управляемые данные, моки и время, раннеры API/браузера, сбор доказательств
Контрактные проверки	Версии контрактов, запуск проверок потребителя и поставщика в доступных границах проекта
Smoke и регрессия	Постоянные команды, состав наборов, события запуска, отчёты и владельцы поддержки
Нагрузка и chaos, если применимы	Выделенный контур либо согласованное окно, изоляция ресурсов, измерения, управление воздействием и восстановление; сохранение работы соседних задач и приёмки

Не все проверки требуют полноценного DEV/UAT. Тестовый раннер со своими временными зависимостями планируется отдельно от пула стендов приложения. Для стендовых проверок сохраняется единая схема сборки, публикации артефакта в хранилище и деплоя.

При отсутствии нужного раннера, данных или средств наблюдения создаётся работа в плане готовности. До её выполнения согласуется временный способ проверки и явно отмечаются ограничения.

Ожидания выделения среды, подготовки ревизии, очереди CI и инфраструктурных отказов измеряются по ЖЦ задач и плану мониторинга. Они отделяются от очереди человека на проверку; занятая среда не считается готовой только по работающему URL. На PoC могут применяться временные среды; различия и проверки целевого контура фиксируются перед Пилотом.

9. Пример: тестовая инфраструктура для трёх принимающих

Вымышленный сервис обработки заявок, независимый от примера «Меридиан Сервис». Названия сред и инструменты условные; выбранные числа иллюстрируют решение проекта, а не обязательный норматив.

Исходные условия

Команда развивает сервис обработки заявок. Три аналитика формулируют и принимают задачи. Агент должен одновременно работать над тремя задачами. Приложение взаимодействует с внешним справочником и сервисом уведомлений; в процессе есть таймер напоминания и закрытие просроченной заявки.

В этом примере подходящих сред для приёмки у команды нет, поэтому новые UAT создаются по образцу DEV. Выбранный состав: **3 DEV + 3 UAT — всего 6 независимых сред**. Это не шесть отдельных серверов по обязательному правилу: способ размещения выбирается под инфраструктуру проекта.

Распределение сред

Среда	Кто работает	Назначение и правило занятости
DEV-1	Агенты	Задача A: новое поле заявки
DEV-2	Агенты	Задача B: повторная попытка обращения к справочнику
DEV-3	Агенты	Задача C: напоминание и закрытие просроченной заявки
UAT-1	Аналитик Анна; агент по её запросам	Приёмка результата, выбранного Анной
UAT-2	Аналитик Борис; агент по его запросам	Приёмка результата, выбранного Борисом
UAT-3	Аналитик Вера; агент по её запросам	Приёмка результата, выбранного Верой

В DEV люди подключаются только при проблеме, которую агенты не смогли решить. UAT закреплены за принимающими; после приёмки они могут использовать их для следующих задач. Ко всем трём UAT у агента есть доступ для разбора замечаний, независимо от исполнителя развёртывания.

Для задачи B процесс сборки публикует артефакт B-17 в хранилище. Агент выполняет деплой B-17 из хранилища в DEV-2 для проверки. Для приёмки Анной выполняется отдельный деплой того же B-17 из хранилища в UAT-1. Оба деплоя ссылаются на один неизменяемый идентификатор артефакта; каждый стенд получает свою конфигурацию и данные.

В этом примере используется параметризованный деплой. Если CI/CD продуктовой команды запускается только из фиксированных веток, допустим вариант со служебными деплойными ветками. Требования к изоляции сред и цепочке «ревизия → сборочный артефакт → деплой» остаются теми же.

Решение по архитектуре

Компонент	Решение для примера	Что переиспользуем или создаём
Сборка и хранилище артефактов	Сборка по точной ревизии, публикация под неизменяемым идентификатором	Используем хранилище пакетов или контейнерных образов; настраиваем публикацию и связь с ревизией

Компонент	Решение для примера	Что переиспользуем или создаём
Деплой приложения	Каждый стенд получает выбранный артефакт из хранилища и свою конфигурацию	Переиспользуем общий кластер; добавляем параметризованный деплой по идентификатору артефакта
Базы и миграции	Отдельная база на среду; миграции и сброс адресуются выбранной базе	Переиспользуем сервер БД при достаточных ресурсах; создаём независимые базы
Очереди и обработчики	Отдельные очереди и обработчики для среды	Расширяем существующие настройки и проверяем отсутствие перекрёстного потребления
Кэш и файлы	Изолированные ключи и хранилища; сброс только целевой среды	Переиспользуем инфраструктуру хранения с отдельными областями данных
Моки интеграций	По отдельному экземпляру сервиса моков на среду; каждый имитирует справочник и уведомления	Создаём один переиспользуемый пакет моков и развёртываем шесть экземпляров
Таймеры и бизнес-время	Управление таймерами процесса плюс виртуальные часы для расчёта срока заявки; настройки отдельны для среды	Создаём инструменты, поскольку исходное приложение зависит от реального времени
Данные и диагностика	Версионированные наборы, сброс, журналы и связь с ревизией приложения	Дорабатываем существующие средства подготовки данных

В этом примере отдельные экземпляры моков выбраны потому, что существующий мок имеет глобальные настройки ошибок и общий журнал. Один физический сервис можно было бы оставить после добавления и подтверждения независимых контекстов для всех шести сред.

Смена задач и состояние БД

Вымышленный пример: задача A переименовала поле `status` в `state`. Следующая задача B основана на исходной версии и ещё использует `status`. Деплой артефакта B поверх оставшейся БД задачи A не даст работоспособную среду.

Для DEV-1 заполнена такая карточка:

Поле	Значение в примере
Цель сброса	Только БД `dev1_app` и связанные состояния DEV-1
Исходный комплект	`base-v12`: снимок со схемой и историей миграций V12; базовые данные D4. Ветки A и B начинались с V12
Целевая ревизия	SHA задачи B; артефакт B-17 опубликован в хранилище
Порядок	Зарезервировать среду, сохранить доказательства A, остановить записи и фоновые задания; восстановить `base-v12`; применить изменения B; загрузить сценарный набор `B-cases-v2`; выполнить деплой B-17 из хранилища и проверки

Поле	Значение в примере
Связанные состояния	Очистить очереди и кэш DEV-1, сбросить её моки и таймеры, установить начальное время набора B
Проверка	В схеме есть ожидаемый B столбец `status`, миграции A отсутствуют, данные и smoke соответствуют B; повтор процедуры даёт тот же результат
Изоляция	DEV-2, DEV-3 и занятые UAT продолжают работу; результаты проверки их состояния приложены к прогону

В журнале сохраняются фактические версии и доказательства, а не только ссылка на эту инструкцию. Если B основана на другой истории, для неё выбирается другой совместимый комплект или создание БД с нуля. Снимок `base-v12` не универсален.

Отдельно для A проверяется **обновление** БД V12 с существующими записями: после переименования значения статусов сохранены в `state`. Восстановление уже изменённой схемы такую проверку не заменяет.

Допустим, в начале PoC автоматических миграций ещё нет. Для изменения A агент готовит версионированный SQL-скрипт с проверками; назначенный разработчик выполняет его по запросу HITL и фиксирует результат. До продолжения агент проверяет фактическую схему и данные. Временный порядок содержит также восстановление при частичном выполнении.

Пробел	Работа плана	Ответственный и срок в примере	Критерий результата
Изменения БД требуют ручного запуска	Внедрить автоматические миграции, перенести исходную схему и скрипты, подключить запуск к деплою	Разработчик; конец фазы 2 PoC. В рабочем плане — имя и дата	Создание с нуля и обновление V12 с данными проходят; выполненные изменения повторно не применяются; сбой останавливает подготовку среды и фиксируется
Сброс и данные готовятся вручную	Подготовить команду восстановления по ID среды и версии набора	ИИ-инженер продуктовой команды; до типичных задач фазы 3	Повторная подготовка даёт заданные схему и данные, соседняя среда не затронута

Эти сроки иллюстрируют план, а не являются общим гейтом методологии. Если автоматизация продолжается на Пилоте, сохраняются проверенный временный порядок, выделенное время исполнителя и актуальный срок работ. Его участие и ожидание фиксируются в показателях.

Для занятой UAT тот же технический рецепт допустим только после разрешённого изменения состояния приёмки. При запрете сброса общего pre-prod команда использует другую среду или проверенное совместимое обновление.

Как выглядит параллельная работа

1. В DEV-1 агент проверяет новое поле на обычных данных.
2. Одновременно в DEV-2 мок справочника отвечает ошибкой на первые два вызова и успехом на третий. Агент проверяет повторные попытки и журнал запросов.
3. В DEV-3 агент готовит заявку, продвигает её бизнес-время и запускает нужный таймер. Проверяет напоминание, переход статуса и отсутствие лишних уведомлений.

4. В UAT-1 Анна продолжает приёмку ранее подготовленной версии. Изменения моков, времени и сбросы DEV не влияют на её среду.

Сначала проверяется согласованная последовательность управления бизнес-временем и таймерами. Простого изменения даты заявки недостаточно: должны выполниться предусмотренные обработчики и связанные проверки срока.

Деплой для приёмки: варианты

Развёртывание поручено агенту

Анна просит агента выполнить деплой артефакта B-17 из хранилища в UAT-1 и загрузить набор «Заявки для проверки справочника». Агент сообщает:

- Адрес UAT-1, идентификатор и ссылку на артефакт B-17 в хранилище, ревизию кода и запись деплоя.
- Версию набора данных и активный сценарий мока.
- Результат проверки готовности и ссылки на проверки артефакта B-17.
- Известные ограничения.

После начала приёмки агент не заменяет эту версию из-за появления нового результата. Анна может отдельно попросить включить отказ справочника, вернуть обычный ответ, загрузить демонстрационные данные или развернуть исправление. Каждая операция выполняется только в UAT-1 и фиксируется; при изменении результата указывается, что проверить повторно.

UAT-2 и UAT-3 продолжают работу независимо.

Принимающий развёртывает сам

Борис выбирает ревизию ветки задачи, запускает сборку и публикацию артефакта либо выбирает уже опубликованный артефакт нужной ревизии. Через привычный процесс команды выполняет его деплой из хранилища в UAT-2. Фиксирует идентификатор артефакта, ревизию и набор данных. Агент не обязан выполнять это развёртывание, но имеет доступ к UAT-2: при замечании Бориса он проверяет описанное поведение и получает необходимые журналы и контекст состояния. Развёртывание исправления снова выполняется выбранным Борисом и командой способом.

Если уже есть подходящий pre-prod

При наличии подходящего pre-prod команда могла бы проводить приёмку там и не создавать новые UAT для покрываемых им сценариев. Деплой опубликованного артефакта из хранилища выполнялся бы штатным процессом продуктовой команды; команда согласовала бы состав сборки, время приёмки и изменения среды. Обязательное условие сохранилось бы: агент имеет доступ к этой среде для разбора замечаний, даже если не может сам развёртывать или сбрасывать её.

Работы подготовки: пример

Возможность	Исходное состояние	Работа	Ответственный	Когда нужна	Подтверждение
3 DEV и 3 UAT	Есть один общий стенд	Подготовить шесть изолированных конфигураций и базы, правила занятости и развёртывание	DevOps	Первая DEV — до фазы 1 PoC; полный состав — до фазы 3	Параллельные деплои артефактов из хранилища, сброс одной среды не меняет остальные
Доступ агента к каждой UAT	Доступ есть только у людей	Обеспечить доступ к фактической версии, поведению, журналам и необходимым данным; согласовать доступные операции	DevOps	До приёмки	Агент разбирает замечание в UAT, развёрнутой Борисом самостоятельно
Сборка и деплой	Существующий процесс нужно проверить	Настроить публикацию артефактов и деплой из хранилища по идентификатору	DevOps	До первой задачи PoC	В каждой записи деплоя видны хранилище, артефакт и ревизия
Управляемые моки	Один мок с глобальным переключателем ошибок	Добавить программные сценарии, сброс и историю; развернуть по экземпляру на среду	Разработчик; ИИ-инженер помогает с настройкой	До задач с интеграциями	Одновременно успех в DEV-1 и ошибки с восстановлением в DEV-2
Таймеры и виртуальное время	Проверки требуют реального ожидания	Добавить управление таймерами и согласованный источник бизнес-времени	Разработчик; архитектор уточняет ограничения	До задачи C, не позднее фазы 3 PoC	Просрочка и напоминание воспроизводятся без ожидания; соседние среды не затронуты
Данные	Наборы загружаются вручную	Подготовить команды загрузки и восстановления по среде	Тестировщик; ИИ-инженер помогает с запуском	До зависимых сценариев и приёмки	Агент дважды воспроизводит сценарий из одинакового набора
Инструкции и диагностика	Знания у одного специалиста	Описать операции, ошибки, сбор журналов и обращение за помощью	ИИ-инженер; DevOps помогает с доступами	При совместном освоении диагностики, до передачи этой работы клиенту	ИИ-инженер продуктовой команды выполняет подготовку по инструкции с согласованной помощью

В конкретных работах трекера вместо ролей указываются имена и даты; у каждой работы один ответственный, вклад соисполнителей задаётся отдельно. Сроки выше показывают зависимости, а не универсальную оценку трудоёмкости.

Если инструменты времени не готовы, команда не объявляет задачу С проверенной. Её либо откладывают с обновлением плана, либо явно исключают из текущей области и сохраняют работу по созданию инструментов для последующего включения.

Что приложить к решению о готовности

- Реестр шести сред: адреса, назначение, владельцы, текущее использование и ссылки на управление.
- Карту изоляции данных, очередей, моков и времени.
- Результаты параллельного прогона, сброса и проверки независимости.
- Выбранный способ подготовки результата в каждой UAT, исполнителя развёртывания и пример передачи контекста приёмки.
- Подтверждённый доступ агента к каждой используемой UAT для диагностики замечаний.
- Оставшиеся ограничения и план их устранения.

Во время Пилота ожидание сред, сбои инструментов и действия по улучшению отражаются в журнале показателей и журнале улучшений.

Пример развития тестового набора

Допустим, на старте у команды нет поддерживаемых автотестов. Это не мешает первой задаче PoC: агент выполняет явные проверки, сохраняет временный скрипт и результаты. Далее команда закрепляет повторяющиеся сценарии по стратегии тестирования.

Сценарий	Подходящая проверка	Где выполняется и как развивается
Правила нового поля заявки	Unit: валидация и граничные значения	Без стенда, в тестовом образе; сохраняется постоянная команда запуска
Сохранение нового поля	Интеграционная проверка реальной БД	Тестовый раннер и временная БД, без полного DEV; проверяются миграция и сохранённые данные
Создание заявки через API	API-автотест	На DEV с артефактом из хранилища, исходными данными и независимыми моками
Ввод и отображение поля	UI-автотест	Браузерный раннер и DEV; основной маршрут включается в регрессию
Повторные вызовы справочника	Проверка отказов через API; контрактная проверка взаимодействия отдельно	Мок задаёт последовательность ошибок и успеха; проверяются результат и число попыток. Совместимость контракта подтверждается доступной проверкой поставщика

Сценарий	Подходящая проверка	Где выполняется и как развивается
Напоминание и просрочка	Unit для расчёта срока плюс интеграционный сценарий времени	Простые правила — без стенда; процесс с таймером — на DEV с управлением временем и наблюдением побочных эффектов
Готовность после деплоя	Smoke из коротких API/UI-сценариев	Запускается для конкретного артефакта; первые сохранённые скрипты превращаются в регулярный набор
Массовое создание заявок	Нагрузка, если это значимый риск	Отдельно планируются ресурсы, данные, профиль и критерии; это не условие старта PoC
Сбой экземпляра обработчика	Контролируемая проверка устойчивости, если нужна	В выделенном контуре проверяются сохранность заявок и восстановление; не затрагивается текущая приёмка

В начале PoC проходит онбординг: команды проверок и деплоя включаются в проектную литературу и проверяются на минимальном пути. После онбординга закрепляются повторяемые проверки простых задач, затем — выбранных целевых сценариев. Сроки фаз определяются проектом. Оставшиеся пробелы получают ответственных и сроки. На Пилоте команда еженедельно рассматривает, какие временные проверки повторяются и что следует добавить в поддерживаемый набор.

Это пример выбора проверок, а не требование дублировать каждый сценарий на всех уровнях.

Примеры механизмов для сложных сценариев

Это примеры для разных продуктов; эмуляторы оборудования и другие инструменты включаются в план только при наличии соответствующих задач.

Сценарий	Подготовленный механизм	Как агент проверяет результат
Случайный выбор исполнителя	Отдельный генератор с seed и заданной последовательностью	Повторяет неуспешный выбор и принудительно проходит граничные варианты
Напоминание по сроку	Начальные часы, продвижение времени и запуск таймера	Проходит границу срока без ожидания; видит одно напоминание в тестовом приёмнике
Ошибка после отправки уведомления	Почтовый приёмник с API и сценарий повторной операции	Проверяет адресата, содержимое и отсутствие второго письма при повторе
Два одновременных изменения заявки	Барьер перед фиксацией изменений и журнал транзакций	Воспроизводит конфликт, проверяет результат по правилам продукта и отсутствие потерянного изменения
Чтение показания контроллера	Эмулятор протокола с профилем, командами и отключением	Подаёт показание, ошибку и восстановление; проверяет обмен без доступа к реальному устройству

Сценарий	Подготовленный механизм	Как агент проверяет результат
Повтор команды после рестарта	Исходные данные, последовательность команд, остановка и восстановление процесса	Подтверждает сохранённое состояние и отсутствие повторного эффекта
Разные локали и feature flags	Два профиля в отдельных процессах	Получает ожидаемое поведение каждого профиля без переноса глобальных настроек

В запись прогона входят выбранные параметры, версии, последовательности и ссылки на доказательства. Отсутствующий механизм получает работу в существующем плане подготовки; список сам по себе готовность не подтверждает.

Стратегия тестирования при внедрении ИИ в разработку ПО

1. Старт без автотестов и развитие покрытия

До первой задачи должен существовать достаточный способ проверить её результат, в том числе временный. Развитие остального тестового покрытия может идти позже; средство проверки, необходимое первой задаче, нельзя откладывать под видом общего плана автоматизации. Способ проверки определяется по ЖЦ задачи и доступной тестовой инфраструктуре.

Наличие готовых автотестов не является условием старта внедрения. Развитие поддерживаемого автоматизированного тестирования — ключевое условие устойчивого эффекта от агента. Оно позволяет повторять проверки без постоянного написания и отладки новых средств проверки.

Без готового набора агенты выполняют аналог «ручного» тестирования: исследуют поведение, вызывают API, работают с UI и создают временные скрипты. Для первых задач это допустимо. Но если каждую следующую задачу проверять заново написанными скриптами, снова расходуются время, инференс и участие людей, а защита от повторных дефектов не накапливается.

Проверка временным скриптом может быть автоматической по способу выполнения, но ещё не является поддерживаемым регрессионным тестом. Для этого её нужно сохранить, проверить, обеспечить повторяемый запуск и включить в регулярный набор.

Наличие тестов само по себе не гарантирует успеха. Важно, чтобы они проверяли нужное поведение, обнаруживали ошибки и использовались на актуальном результате. Успешное завершение Пилота определяется достижением согласованных показателей и выполнением остальных условий чек-листа завершения. Сам по себе план улучшений не означает успешного завершения.

2. Какие тесты должны появиться в проекте

В целевом наборе предусматриваются **unit-тесты, API-автотесты и UI-автотесты для применимых частей продукта**. Кроме них рассматриваются интеграционные и контрактные проверки. Для каждого применимого вида фиксируется, что уже есть, какие пробелы выявлены и как проверяется выбранная задача. Отсутствие UI или API — обоснованная неприменимость соответствующего вида, а не требование создавать искусственные проверки.

Термины описывают разные стороны теста: unit и интеграционный — границу проверки, API и UI — интерфейс взаимодействия, E2E — сквозной охват. Например, API-тест может быть интеграционным или сквозным. В реестре это одна проверка с несколькими признаками, а не несколько тестов.

Вид	Что проверяет	Где и с чем запускается	Что предусмотреть
Unit	Небольшие единицы логики: расчёты, правила, переходы состояния	Без развёрнутого стенда, в процессе тестового раннера; зависимости изолированы	Быстрый запуск, граничные случаи, фиксация исправленных ошибок

Вид	Что проверяет	Где и с чем запускается	Что предусмотреть
Компонентные и интеграционные	Работу компонента и взаимодействия с БД, очередью, файловым хранилищем или другим модулем	В тестовом окружении с нужными зависимостями; полный DEV-стенд часто не нужен	Подготовка и очистка состояния, миграции, реальные особенности используемых зависимостей
Контрактные	Совместимость запросов, ответов и сообщений между потребителем и поставщиком	Отдельными проверками сторон контракта; обычно без полного сквозного стенда	Проверяемые ожидания и версии контрактов; проверка стороны поставщика, а не только собственного мока
API-автотесты	Поведение через API: бизнес-сценарии, валидацию, доступ, ошибки	Для проверок развёрнутого приложения — DEV с заданными данными и зависимостями; для компонента возможен отдельный тестовый запуск	Положительные, отрицательные и граничные сценарии, состояние после операции
UI-автотесты	Поведение пользовательского интерфейса: действия, отображение, взаимодействие элементов	Компонент UI можно проверить изолированно; браузерные проверки приложения — на DEV с управляемыми данными	Основные пользовательские маршруты и риски UI, устойчивый способ обращения к элементам
Сквозные E2E	Полный бизнес-сценарий через несколько частей системы	На подготовленном стенде; реальные зависимости и заменённые моками явно перечислены	Набор критичных маршрутов; сложная среда не должна быть единственным способом проверить простое правило

Быстрые локальные проверки и проверки взаимодействий дополняют сквозные сценарии; распределение зависит от архитектуры, универсальное соотношение количества тестов не задаётся. Описание уровней и практики тестовой пирамиды.

Контрактные проверки помогают обнаруживать несовместимость взаимодействующих сторон. Тест только на собственном моке не доказывает, что реальный поставщик соблюдает ожидания. Описание подхода в документации Pact. Использование именно Pact не требуется.

Unit-тесты и другие проверки без стенда

«Без стенда» означает, что для прогона не требуется деплой приложения на DEV или UAT. Тесты могут выполняться в специальном контейнерном образе с нужными библиотеками и инструментами. Образ запуска и версии зависимостей фиксируются.

Для части проверок может понадобиться отдельная тестовая БД, контейнер БД или временная схема. По принятой здесь терминологии проверка реального взаимодействия с такой БД относится к интеграционным тестам, даже если у продуктовой команды весь набор называется unit. Это не мешает запускать их вместе в CI; важно понимать зависимости и стоимость запуска.

Для тестовой БД задаются изоляция параллельных прогонов, начальные данные, миграции и очистка. Её состояние не зависит от текущей занятости DEV/UAT. Если используется упрощённая замена БД, отдельно отмечается, какие особенности реальной БД таким прогоном не проверяются.

3. Назначение проверок: что ещё нужно учесть

Это не дополнительные взаимоисключающие уровни. Smoke и регрессия могут состоять из API-, UI- и других тестов; проверки отказов также могут выполняться на разных уровнях.

Назначение	Содержание	Применение в методологии
Smoke	Короткая проверка готовности и основных функций	После деплоя артефакта из хранилища на выбранный стенд; на старте допустим явный проверяемый сценарий, далее — поддерживаемый автоматический набор
Регрессия	Сохранность ранее работавшего поведения	Затронутые проверки — по задаче; более широкий набор — по согласованному расписанию или перед выпуском
Отказы и восстановление	Ошибки, тайм-ауты, повторы, недоступность зависимости, восстановление и целостность данных	Управляемые моки и тестовые данные для значимых сценариев; не только проверка появления сообщения об ошибке
Регулярные операции и время	Сроки, расписания, таймеры, повторные и пропущенные срабатывания	Виртуальное время или управление заданиями по описанию инфраструктуры; проверяются результат и побочные эффекты
Данные и миграции	Изменения схемы, преобразование и сохранность данных	При соответствующих изменениях; воспроизводимый набор исходных данных и проверка результата миграции
Нагрузка и производительность	Поведение при выбранном профиле нагрузки, росте объёма данных и длительности работы	По рискам и согласованным требованиям; отдельный план условий, измерений и ресурсов
Устойчивость / chaos	Поведение системы при управляемом нарушении работы компонентов или сети	Контролируемый эксперимент с гипотезой, наблюдением и восстановлением; применяется там, где оправдан рисками
Безопасность	Права доступа, изоляция данных, значимые негативные сценарии; проверки зависимостей и кода	Обязательные правила продуктовой команды сохраняются; объём специальных проверок согласуется для продукта
Совместимость и доступность UI	Поддерживаемые браузеры и устройства, работа с клавиатурой и другие требования доступности	Если эти свойства значимы для выбранного интерфейса
Исследовательское тестирование и UAT	Новые риски, удобство, смысл бизнес-поведения и продуктовая приёмка	Люди сохраняют эти функции; агенты и автотесты помогают, но не подменяют решение принимающего

Статический анализ, линтеры, проверка типов и анализ зависимостей также входят в проверки качества, но не заменяют выполнение тестов поведения.

Нагрузочные проверки

Выбирается конкретная цель: обычная ожидаемая нагрузка, повышенная нагрузка (stress), резкий всплеск (spike) или длительная работа (soak). Эти виды отвечают на разные вопросы; перечень и профиль определяются рисками проекта. Описание типов нагрузки в документации k6.

Для прогона фиксируются артефакт из хранилища, конфигурация и ресурсы стенда, объём данных, нагрузочный профиль и измеряемые показатели: например, задержки, доля ошибок и пропускная способность. Пороговые значения задаёт команда по требованиям, а не по результату уже выполненного прогона.

Нагрузку выполняют на выделенном контуре или в согласованном окне с достаточной изоляцией. Влияние генератора нагрузки, общих ресурсов и моков учитывают при интерпретации. Ускорение бизнес-времени не заменяет длительный прогон для поиска утечек ресурсов или деградации. Без репрезентативных условий выводы о производительности ограничены.

Проверки отказов и chaos

Детерминированный ответ мока «ошибка», после которого проверяется повторная попытка, — проверка обработки отказа. Chaos-эксперимент рассматривает поведение системы при управляемом воздействии: например, остановке экземпляра сервиса или нарушении сетевого взаимодействия. Его основа — гипотеза о наблюдаемом поведении и попытка её проверить. Principles of Chaos Engineering.

В рамках внедрения такой эксперимент планируется на согласованном тестовом контуре: задаются воздействие, границы, ожидаемое поведение, измерения, условия остановки и восстановление. Эксперимент не должен нарушать чужие задачи или текущую приёмку.

Связь с границами внедрения

В базовый поток по-прежнему входят функциональные задачи, выбранные для PoC и Пилота. Описанные виды проверки нужны для качества этих задач и развития тестового набора. Это не означает автоматического включения самостоятельного потока реализации нефункциональных требований.

Существующие обязательные проверки продуктовой команды сохраняются. Новая программа нагрузки, chaos или других специализированных проверок включается по применимости и риску с отдельными ресурсами и планом. Если такой проверки нет, а она нужна для текущей задачи, пробел фиксируется вместе со способом его закрытия; неподтверждённое свойство не объявляется проверенным.

4. Как развивать тесты по этапам

Этап	Что делаем
Подготовка	Инвентаризируем существующие тесты, команды запуска и зависимости. Для выбранного потока определяем применимые виды, пробелы, ответственных и план. Отсутствие готового набора не блокирует старт само по себе

Этап	Что делаем
PoC: онбординг	ИИ-инженер настраивает команды и навыки проверок, проходит минимальный сценарий, сохраняет ревизии и результаты; применяет готовые тесты или достаточный временный способ
PoC: простые задачи	Закрепляются повторяющиеся unit-, API- и UI-проверки по применимости; готовятся данные и воспроизводимый запуск
PoC: целевые задачи	Проверяются выбранные типы задач, возврат замечаний, повторное выполнение и достаточность доказательств; уточняются границы Пилота
Пилот	Команда выполняет рабочие проверки; ИИ-инженер помогает со сложными случаями. По затратам, возвратам и дефектам выбираются улучшения тестов и процесса

Полный набор всех видов к началу Пилота не требуется. Однако для каждой выполняемой задачи должен быть согласован достаточный способ проверки результата. Отсутствие готовых автотестов не означает отсутствие тестирования и не отменяет обязательных проверок продуктовой команды.

Приоритет автоматизации: повторяющиеся критичные сценарии, проверки изменяемой логики, случаи уже найденных дефектов и проверки, которые отнимают заметное время на повторную подготовку. Процент покрытия кода и число тестов не используются как единственные цели.

5. Когда временный скрипт становится поддерживаемым тестом

Для повторяющейся проверки выбирается работа по включению в поддерживаемый набор. Готовность подтверждается следующим:

- Тест сохранён в репозитории и связан с проверяемым требованием, сценарием или дефектом.
- Ожидаемый результат определён по требованиям; тестировщик или разработчик проверил, что тест способен обнаружить нужную ошибку.
- Есть постоянная команда запуска, зафиксированные зависимости, подготовка и очистка данных.
- Параллельные прогоны не меняют состояние друг друга.
- Тест запускается регулярно в CI или другом согласованном процессе, а его отказ виден ответственному.
- Результаты содержат ревизию тестов, проверяемую ревизию кода, условия и доказательства; для стендовых проверок — идентификатор артефакта и запись деплоя.
- Назначен владелец поддержки и порядок изменения теста вместе с продуктом.

Агент может писать и поддерживать тесты. Разработчик рассматривает техническую корректность, тестировщик — смысл сценариев и достаточность проверок; роли могут совмещаться. Подробный маршрут описан в правилах Git.

Нестабильный тест разбирают как проблему: сохраняют все результаты повторных запусков, назначают исправление, явно отмечают исключения. Повтор до первого зелёного результата или ослабление утверждений не подтверждают качество.

6. Запуск и доказательства

Без стенда: тестовый раннер проверяет точную ревизию кода в заданном окружении; отдельно поднимаются необходимые временные зависимости. Это не деплой приложения в DEV/UAT.

На стенде: приложение собирается, артефакт публикуется в хранилище, затем выполняется деплой выбранного артефакта на целевой стенд. Тесты фиксируют именно этот артефакт, конфигурацию, данные, сценарии моков и настройки времени. UAT используется по правилам приёмки продуктовой команды; агент имеет доступ для разбора замечаний, а автоматические операции выполняет в согласованных пределах.

Проверки для каждой задачи выбираются по изменению и риску. Быстрые проверки запускаются при изменении кода, smoke — после деплоя, затронутая регрессия — после исправлений, более широкие и дорогие прогоны — по установленным событиям или расписанию. Конкретные команды и порядок входят в профиль проекта.

7. Покрытие и улучшения тестирования

Состав проверок и команды запуска хранятся с тестами и в инструкциях продукта. При выборе задач уточняются применимые виды, доступные проверки и конкретные пробелы. Не требуется заранее составить полный план всех недостающих артефактов.

Таблица ниже помогает разобрать текущую область, но не является отдельным обязательным реестром. Выбранные работы назначаются в трекере; выполненные изменения тестов, инструментов и процесса сохраняются в журнале улучшений с исполнителями и результатом проверки.

Вид / назначение	Применимость и сценарии	Что уже есть и где хранится	Как и где запускается	Пробел или временный способ	Что добавляем, кто отвечает и срок	Как подтвердим результат
Unit	…	…	Без стенда; образ и команда	…	…	…
Интеграционные / контрактные	…	…	Зависимости, БД, контракт	…	…	…
API	…	…	Стенд или компонент; команда	…	…	…
UI / E2E	…	…	Браузер, стенд, данные	…	…	…
Smoke / регрессия	…	…	Состав набора и событие запуска	…	…	…
Время / отказы / миграции	…	…	Инструменты, сценарии	…	…	…

Вид / назначение	Применимость и сценарии	Что уже есть и где хранится	Как и где запускается	Пробел или временный способ	Что добавляем, кто отвечает и срок	Как подтвердим результат
Нагрузка / chaos / другие специальные проверки	…	…	Контур, условия, ресурсы	…	…	…

Один тест может входить в несколько наборов; его сопровождение и трудозатраты не учитываются повторно.

На еженедельном разборе Пилота фиксируются:

- Какие значимые сценарии теперь проверяются повторяемым набором и какие остаются непокрытыми.
- Где агенты снова писали временные скрипты, сколько это стоило и что решено сохранить.
- Затраты на создание, поддержку и запуск тестов; время прогонов и ожидания.
- Нестабильные тесты, пропуски и дефекты, не обнаруженные текущими проверками.
- Изменение трудозатрат и времени задач с учётом качества.

Это диагностика причин эффекта, а не новый личный рейтинг или дополнительное основание завершения Пилота. Привязка к учёту — в метриках, показатели — в журнале показателей, выполненные улучшения — в журнале улучшений.

На PoC регулярные проверки и возврат замечаний агентам ведёт ИИ-инженер. Специалисты продукта обучаются и наблюдают работу, позднее могут участвовать под его сопровождением; самостоятельная работа на PoC не обязательна. При временных инструментах и средах сохраняется смысл контактов и проверок будущего Пилота. Для Пилота действуют правила приёмки в ЖЦ задач.

НА ПРЕДПРИЯТИИ

Паспорт программы внедрения

Паспорт описывает внедрение агентской разработки на предприятии: зачем оно проводится, какие продукты охватывает, как организованы волны, кто принимает решения и какие ресурсы выделяются. **Состав команды программы включён в паспорт**; отдельная карточка команды для этих же сведений не нужна.

Пример заполнения: Меридиан Сервис.

Шаблон

Скопируйте раздел ниже в рабочий документ программы. Пояснения замените сведениями проекта. Если решение не принято, так и запишите; не представляйте предварительные значения утверждёнными.

Паспорт программы «…»

1. Программа и цель

Предприятие: …

Состояние на дату / версия: …

Горизонт программы: …

Цель внедрения: …

Исходная проблема и основания: …

Ожидаемый результат и согласованные ориентиры: …

2. Охват

Продукты и общие сервисы: …

Первая волна и основания выбора: …

Что исключено или требует отдельного решения: …

3. Конфигурация и общие условия

Способ организации внедрения: …

Что обеспечивает предприятие и что выполняет команда внедрения: …

Общие условия платформы, инференса, инфраструктуры и доступа: …

Корпоративные требования и ограничения: …

4. Календарь программы

Период	Этап / волна и охват	Ожидаемый результат
…	…	…

Паспорта подключаемых продуктов: …

5. Команда программы

Роль / функция	Участник	Принадлежность	Выделенное время / период
…	…	…	…

Совмещения и ограничения суммарной загрузки: …

Порядок взаимодействия и рассмотрения общих вопросов: …

6. Ресурсы и ограничения

Выделенные ресурсы и бюджетные ограничения: …

Основания для пересмотра состава волн, сроков и ресурсов: …

7. Открытые решения и подтверждения

Что ещё не определено: …

Что согласовано, кем и на каком основании: …

Изменения относительно предыдущей версии: …

Связанные документы и источники: …

Как заполнять

Паспорт составляют при оформлении программы и уточняют по результатам волн. Для первоначальных ограниченных проб до решения о программе заполнение паспорта не является условием начала работы. Подготовку документа организует руководитель программы; цели, границы и выделение ресурсов подтверждаются уполномоченными участниками предприятия.

- **Цель и результат.** Запишите, какое изменение производства ПО требуется предприятию и почему оно нужно. Различайте эффект внедрения и бизнес-показатели самих продуктов. Ориентиры ускорения и затрат должны иметь основания; не подставляйте универсальный процент. Неопределённое значение обозначайте прямо.

- **Охват.** Перечислите продукты, общие сервисы и существенные исключения. Укажите первую волну и основания выбора. Доступность компонента через API не означает права менять его код или передавать агентам работы подрядчика.
- **Конфигурация.** Укажите самостоятельное внедрение или участие выделенной команды. Отделите общие решения предприятия от настройки каждого продукта. Размещение и инфраструктуру описывайте кратко; технические схемы и перечни доступов остаются в соответствующих документах.
- **Календарь.** Дайте горизонт программы и волны с ожидаемыми результатами. Локальные планы всех продуктов здесь не повторяются. Плановая дата не подтверждает выполнение условий перехода; Подведение итогов подробно на работы не раскладывается.
- **Команда и ресурсы.** Для каждого участника укажите роль или функцию, человека, принадлежность и выделенное время. При совмещении функций не складывайте один резерв времени повторно. Общая координация не делает руководителя программы ответственным за каждую работу продукта.
- **Решения.** Различайте запланированное, выделенное и подтверждённое. Записывайте открытые вопросы и ссылки на основания изменений. Дата в паспорте и заполненная таблица сами по себе не являются согласованием.

Для каждого продукта используется паспорт проекта внедрения. Общие требования и полномочия программы задаются здесь, локальные участники и календарь — в паспорте продукта. Одни и те же люди могут участвовать в нескольких продуктах; их суммарная доступность проверяется на уровне программы.

Паспорт программы «Агентская разработка в Меридиан Сервис»

Учебный пример по шаблону паспорта программы. Отражает плановые условия начала программы из базы Меридиана; не является протоколом утверждения или отчётом о завершённых этапах.

1. Программа и цель

Предприятие: Меридиан Сервис, обслуживание систем вентиляции и холодоснабжения магазинов и распределительных центров.

Состояние на дату / версия: план запуска с 18 мая 2026 года; исходное состояние предприятия — 4 мая, показатели за апрель. На дату исходного среза Джина не развёрнута, решения о начале программы ещё не оформлены.

Горизонт программы: май 2026 — декабрь 2027.

Цель внедрения: кратно ускорить поставку функциональности без сопоставимого роста полной стоимости разработки, сохраняя требуемое качество.

Исходная проблема и основания: команды имеют бэклог на несколько месяцев и не успевают реализовывать изменения с необходимым бизнесу темпом. В ИТ и продуктовом развитии 42 штатных сотрудника; расширение штата на 2026 год не предусмотрено. Исторические показатели команд различаются по дроблению задач и учёту ожиданий, поэтому их нельзя считать готовой сопоставимой оценкой производительности.

Ожидаемый результат и согласованные ориентиры: автономная агентская разработка применяется во всех четырёх продуктах; проверены изменение скорости поставки, полные затраты и ограничения способа работы. Полная стоимость включает труд сотрудников и команды внедрения, платформу, инференс и тестовые среды. Числовые ориентиры ускорения и допустимого роста стоимости в исходных условиях не заданы; подтверждений их согласования нет. Доля заявок через кабинет — бизнес-показатель продукта, а не измерение эффекта программы.

2. Охват

Продукты и общие сервисы: личный кабинет заказчика; планирование выездов; мобильное приложение инженера; склад и запчасти. Общая платформа обеспечивает инфраструктуру всех внедрений и не считается пятым продуктом для отдельного пилота.

Первая волна и основания выбора: личный кабинет. Основной характер работ — разработка функциональности: собственные web/API-репозитории, команда и действующий конвейер и очередь фич на несколько месяцев. Для агентов выбираются обеспеченные участки и независимые фичи с запасом времени на повторные проверки. Сценарий смет с внешними согласованиями не передаётся агентам целиком как первая работа.

Что исключено или требует отдельного решения: закрытая конфигурация 1С и полномочия подрядчика не включаются автоматически. Срочные исправления не превращаются в основной поток пилота. Успех первого Пилота не означает автоматического допуска остальных продуктов или отмены человеческих проверок.

3. Конфигурация и общие условия

Способ организации внедрения: с привлечением специализированной компании «Практика ИИ», которая выделяет команду внедрения. Компания имеет компетенции разработки ПО, настройки агентских платформ и организации производственных процессов; специалисты предприятия сохраняют предметные решения и осваивают работу с агентами.

Что обеспечивает предприятие и что выполняет команда внедрения: предприятие выделяет время людей, ресурсы, связность и доступы. Команда внедрения организует свои работы, устанавливает и адаптирует платформу в предоставленном контуре, обучает специалистов и помогает при затруднениях. Точный состав первого внедрения приведён в паспорте кабинета.

Общие условия платформы, инференса, инфраструктуры и доступа: общий экземпляр `gina-meridian` размещается в новом отдельном Kubernetes-кластере предприятия, инференс GLM 5.3 и GLM 5.3 flash предоставляет «Практика ИИ». Материалы передаются поставщику только при работе агентов с LLM. С PoC используются целевые системы конвейера. Для первого продукта создаются feature-среды в существующем продуктовом кластере; карточка инфраструктуры первого внедрения раскрывает подключения и назначения. На исходную дату установка и агентские подключения ещё не выполнены.

Корпоративные требования и ограничения: подключение систем и обработка данных подчиняются действующим политикам предприятия. Рассмотрение исключений происходит установленным корпоративным порядком вне проекта и может изменить календарь. Установка платформы специалистом внедрения не даёт ему права выделять мощности или менять доступы по своему решению.

4. Календарь программы

Период	Этап / волна и охват	Ожидаемый результат
С 18 мая 2026; первоначальные решения до 29 мая	Начало программы, прогнозирование эффектов	Обоснование цели и ресурсов, выбор первого продукта и предварительной конфигурации
Июнь–октябрь 2026	Пилотная волна: личный кабинет	Подготовка, PoC и Пилот; проверка результата и ограничений
Ноябрь–декабрь 2026	Разбор пилотной волны	Уточнение условий переноса и возможности подключения остальных продуктов
2027 год	Одна волна масштабирования: планирование, склад, мобильный продукт	Внедрения по готовности и возможностям сопровождения; в каждом Подготовка, PoC и Пилот
Декабрь 2027	Подведение итогов	Итоги по продуктам, общим затратам и ограничениям, дальнейшая поддержка

Паспорта подключаемых продуктов: личный кабинет заказчика. Остальные внедрения входят в охват 2027 года, но подробные сроки и назначения в исходных данных ещё не определены. Отдельной волны 2028 года нет.

5. Команда программы

Роль / функция	Участник	Принадлежность	Выделенное время / период
Спонсор	Виктор Мельников	Предприятие, генеральный директор	До 1 ч/нед. и участие в ключевых решениях
Руководитель программы	Алексей Орлов	Предприятие, технический директор	4 ч/нед.
Координация программы; РП первого внедрения	Марина Фомина	Предприятие, руководитель поставки	8 ч/нед. суммарно на запуске и первом внедрении
Методолог после обучения	Елена Миронова	Предприятие, руководитель продуктов	4 ч/нед.
Обеспечение инфраструктурных работ	Евгений Сафонов	Предприятие, руководитель платформы	2 ч/нед., с пересмотром при подключении продуктов

Совмещения и ограничения суммарной загрузки: восемь часов Марины включают обе функции, не складываются в шестнадцать. Назначения не создают новые штатные единицы. На запуске отдельное подразделение офиса не создаётся; постоянная организация и загрузка уточняются к масштабированию. Время на следующие продукты не считается выделенным по аналогии с кабинетом.

Порядок взаимодействия и рассмотрения общих вопросов: роли принимающих решения и координирующих участников определены выше. Расписание встреч и рабочий канал программы в исходных данных не заданы. Корпоративное рассмотрение условий доступа выполняется существующим порядком, без отдельной проектной роли ИБ.

6. Ресурсы и ограничения

Выделенные ресурсы и бюджетные ограничения: в сценарии заданы резервы времени участников предприятия и профессиональной команды для первого внедрения. Это плановые значения; подтверждения фактического выделения и финансовый бюджет не представлены. Состав команды первого проекта не является готовым ресурсным планом всей программы до конца 2027 года.

Основания для пересмотра состава волн, сроков и ресурсов: результаты первого Пилота, доступность инфраструктуры и сопровождения, внешние зависимости, невыполненные условия перехода. Продуктовые договорные обязательства не сдвигаются автоматически ради внедрения. Предполагаемое ускорение агентов не включается в исходную ёмкость команд.

7. Открытые решения и подтверждения

Что ещё не определено: числовые ориентиры ускорения и стоимости, финансовый бюджет, договорные условия команды внедрения, версия поставки, размеры ресурсов и квоты, условия хранения LLM-запросов и расписание взаимодействия программы. Для продуктов 2027 года не заданы подробные сроки и персональные резервы.

Что согласовано, кем и на каком основании: база задаёт плановый сценарий. Проектные протоколы, утверждения и подтверждения выделения ресурсов не представлены; этот пример не доказывает прохождение гейта начала программы.

Изменения относительно предыдущей версии: предыдущая версия паспорта в базе отсутствует.

Связанные документы и источники: описание предприятия, сценарий внедрения, паспорт кабинета. Факты примера берутся из этих материалов; новые решения и подписи за участников не добавлены.

Правила работы в Git

Карточка сведений описывает ветки, MR, проверки, интеграцию и выпуск для общего процесса людей и агентов. Состояния задач находятся в ЖЦ задач, системы и права — в карточке инфраструктуры. Здесь задаются разрешённые операции и события Git/CI, по которым можно восстановить движение конкретной ревизии и ожидания.

Пример: кабинет «Меридиан Сервис».

Шаблон

1. Область и различия этапов

Продукт, репозитории и назначение: … . **Связанные документы:** ЖЦ задач …, инфраструктура …, паспорт … . **Редакция, дата применения, согласование:** … .

Правило	Действующий процесс	PoC	Пилот
Git-система и репозитории	…	…	…
Ветки и целевая ветка MR	…	…	…
Проверки, ревью, права merge	…	…	…
Среды, интеграция и выпуск	…	…	…
Проверка различий	—	Что воспроизводится временно …	Что повторно проверяется в целевых системах …

2. Ветки и полномочия

Ветка / шаблон имени	Откуда создаётся и куда интегрируется	Кто создаёт, изменяет и сливает; запреты	Связь с задачей, средой и ревизией
Основная	…	…	…
Рабочая	…	…	…
Интеграционная / служебная, если применяется	…	…	…
Выпуск / hotfix, если применяется	…	…	…

Защита и конкурентные изменения: обязательные approvals и проверки, запрет обхода защиты и force-push, порядок обновления от базы, конфликтов и совместимых изменений нескольких репозиториев … . Удаление или сброс служебной ветки допускается только в названной области с сохранением необходимой истории. Права агента проверяются технически, не только текстом инструкции.

3. MR, HITL и исправления

Событие	Условие и действие	Кто действует на PoC / Пилоте	Подтверждение и связь с ЖЦ задачи
Draft MR	Незавершённая работа, формальное ревью ещё не запрошено …	…	…
Готов к ревью	Комплект изменений, проверки и материалы готовы; запрос доставлен …	…	…
Начато ревью	Проверяющий приступил к содержательной проверке …	…	…
Требуются изменения	Сохранены замечания и проверенная ревизия …	…	…
Повторно готов к ревью	Новая ревизия, ответы на замечания и нужные проверки …	…	Новый раунд, без сброса предыдущей истории …
Одобрено, ожидает merge	Все обязательные решения получены …	…	…
Выполнен merge / закрыт без merge	Зафиксирован исход, SHA и дальнейшие действия …	…	…

Материалы MR: задача и реализованное поведение; границы изменений; SHA, результаты CI и ссылки на тесты; нужные данные и среда; ограничения, миграции и риски; порядок проверки … .

Любой запрос HITL система отправляет в чат с событием и ссылками на MR, задачу и проверяемую ревизию. Для нескольких обязательных проверяющих запросы различаются по адресату и предмету. Кто может подменять проверяющего и как сохраняется решение: … . Уточнения и замечания из личного разговора сохраняются в доступной агенту записи.

Новая ревизия во время ревью: правила актуальности approvals и CI, повторного запроса и отмены устаревшей проверки … . Старый approval не переносится автоматически на непроверенный код. Закрытие MR без merge и передача человеку сохраняются как исход, а не успешное ревью/поставка.

4. CI, среды и выпуск

Обязательные проверки и момент запуска: build …, тесты …, анализ …; какие проверки блокируют merge, какие выполняются после интеграции … . Исключения и решения сохраняются явно.

Развёртывание: кто выбирает ревизию, запускает CI и резервирует среду, как подтверждаются digest/SHA, готовность и набор данных … . Укажите защиту от параллельных деплоев и запоздалого pipeline. Проверка старого образа не подтверждает новую ревизию.

Интеграция и поставка: условия продвижения между ветками/средами, совместимость нескольких репозиториев, приёмка, разрешение production, фиксация выпущенной версии … . Merge, успешный тест и выпуск — разные события. **Откат и миграции:** … .

5. События для мониторинга

Интервал	Начало → конец	Источник подтверждения
Очередь ревью HITL	Готовность ревизии и доставка запроса → начало содержательного ревью	MR + чат + подтверждение начала работы …
Проверка и решение	Начало ревью → вердикт по ревизии	Review / discussions / approval с проверенным смыслом …
Ожидание merge	Все обязательные approvals и проверки актуальны → merge	Git и CI …
Очередь CI	Готовая к запуску job принята в очередь → началось исполнение	Pipeline/job, зависимости и runner …
Ожидание тестовой среды	Потребность в среде нужной ревизии → среда предоставлена и готова	CI, резервирование, health check, SHA/digest …
Отказ Git, CI, registry или среды	Отказ блокирует действие → подтверждено восстановление	Ошибка и проверка повторного действия …

Метрики определяются в каталоге расчётов, выбор и источники — в плане мониторинга. Общие причины, идентификаторы ожиданий и правила перекрытий задаются в ЖЦ задач. Эта таблица уточняет их события в Git/CI, не создаёт второй учёт тех же интервалов.

Как заполнять

Сначала восстановите существующий маршрут изменения от рабочей ветки до production, включая фактические проверки и полномочия. Свяжите его с переходами трекера. Разработчик описывает правила Git, DevOps — CI и развёртывание, ИИ-инженер — выполнение агентами и доступность событий; конкретные исполнители назначаются в паспорте и работах.

На PoC укажите реально используемые репозитории, ветки и среды. Если они временные, покажите соответствие целевому процессу Пилота: готовность MR, возврат замечаний, повторное предъявление, интеграция и тестирование. ИИ-инженер может выполнять HITL вместо специалистов, а позднее проверять материалы до и после их участия. Смена инструмента не доказывает, что целевой маршрут работает: перед Пилотом повторите проверки затронутых операций и доступов.

Не принимайте создание MR за начало ожидания ревью, первый комментарий — за точное начало работы, а зелёный pipeline — за выполненное тестирование всей задачи. Если нужное событие недоступно, сначала определите достоверный способ его получения. Время до первого комментария допустимо как отдельный показатель, но не как скрытая подмена очереди ревью. Для всех ожиданий показывайте также открытые запросы, их возраст и полноту данных.

Правила Git: кабинет «Меридиан Сервис»

Пример Git-карточки для двух репозиториев кабинета. Основание — текущее описание продукта, связанные правила — ЖЦ задач. Настройки работы агентов ниже предлагаются для PoC и Пилота; их применение ещё предстоит проверить.

1. Область и различия этапов

`customer-portal-web` — React и браузерные проверки; `customer-portal-api` — API, PostgreSQL, потребитель событий и фоновые операции. GitLab, CI и registry входят в карточку инфраструктуры. Точные адреса и технические учётные записи берутся оттуда, не дублируются в этой форме.

Правило	Действующий процесс	PoC	Пилот
Git и ветки	Продуктовый GitLab, защищены main и integration	Планируются те же репозитории и защита	Тот же маршрут с проверенными доступами агентов
Проверяющие	Разработчики и тестировщики продукта	Регулярные проверки ведёт Никита, предметные решения — у продукта	Проверки ведут специалисты продукта; Никита помогает
Среды	Одна общая integration, без feature-сред	Feature-среды готовятся как работа внедрения, до проверки используется доступная общая среда	Запланированы проверенные feature-среды и общая integration
Выпуск	Только после приёмки и человеческого разрешения	Пробы не дают агентам самостоятельного права на production	Те же полномочия; изменение требует отдельного решения

Если PoC временно переносится в отдельный Git или использует упрощённые ветки, записываются фактические репозитории и сопоставление шагов: готовность MR, замечания, повторное ревью, merge, тестирование. Перед Пилотом повторно проверяются изменившиеся права и операции. Временный порядок не заменяет правила продуктовых репозиториев.

2. Ветки и полномочия

Ветка	Создание и назначение	Полномочия и подтверждения
main	Целевая ветка MR, принятый код	Защищена; прямые изменения агента запрещены. Merge выполняет уполномоченный разработчик после проверок и одобрений
feature/PORTAL-… и fix/PORTAL-…	От актуальной main; одна задача, при необходимости отдельная ветка в каждом репозитории	Агент изменяет свои рабочие ветки, открывает MR; в задаче сохраняются все связанные MR

Ветка	Создание и назначение	Полномочия и подтверждения
integration	Согласованная ревизия из main для общей проверки	Защищена; продвижение запускает автоматический деплой CI. Исполнитель продвижения назначается по действующим полномочиям команды
hotfix/*	От действующей версии production	Срочные исправления по существующему процессу; не включаются в агентский поток автоматически

Сергей подтверждает порядок зависимых изменений контрактов web/API. Обновление базы рабочей ветки и разрешение конфликтов не отменяют повторных проверок. Force-push и обход защиты агента в защищённых ветках запрещены. Политика актуальности approvals при новом commit должна быть проверена в GitLab до Пилота.

3. MR, HITL и исправления

Событие	Действие и подтверждение	PoC / Пилот
Draft	Можно показать незавершённое изменение для обсуждения; ожидание обязательного ревью не начинается	Автор MR, Никита сопровождает агента
Готов к ревью	Готов комплект изменений и обязательные проверки; в чат доставлен запрос со ссылкой на MR, Jira и SHA	Никита проверяет готовность на PoC; на Пилоте это выполняется в общем процессе
Начато ревью	Проверяющий начал содержательное рассмотрение; точное событие и его доступность проверяются на PoC	Никита / назначенный разработчик продукта
Требуются изменения	Вердикт и конкретные замечания связаны с ревизией; задача возвращается в «В работе»	Проверяющий, инженер проверяет обратную связь при посредничестве
Исправления готовы	Агент отвечает на замечания, предъявляет новую ревизию и результаты проверок; новый запрос начинает следующий раунд	Никита / штатный проверяющий
Одобрено	Все обязательные решения и CI актуальны для предъявленного кода	Уполномоченные проверяющие по правилам репозитория
Merge / закрытие	Сохраняется итоговый SHA либо исход без merge; при передаче человеку сохраняются изменения и история	Уполномоченный разработчик; агент не получает право merge автоматически

API проверяют Сергей и Игорь; типовые изменения может проверять Денис. Frontend проверяют Елена и Денис. Проверку автора не считают независимым одобрением. На PoC инженер выполняет регулярное рассмотрение материалов, но его проверка не отменяет действующие требования

продуктового репозитория. При позднем подключении специалиста Никита проверяет материалы до его рассмотрения и обратную связь перед возвратом агенту; два обращения учитываются раздельно.

MR содержит результат и границы задачи, Jira-ссылку, текущий SHA, проверки и ограничения, сведения о миграциях и способе проверки. При нескольких MR готовность задачи определяется полным необходимым комплектом. После нового commit результаты старой ревизии не считаются доказательством проверки новой; незавершённый раунд помечается заменённым, а не успешно завершённым. Напоминание о прежнем запросе не открывает новый раунд.

HITL направляется через Mattermost с наступившим событием и ссылками. Предметные ответы и замечания должны быть доступны агентам. Канал, получатели и резервные участники уточняются по паспорту и правилам коммуникаций. Снятый запрос не считается нулевым ожиданием или выполненным ревью.

4. CI, среды и выпуск

До merge обязательны сборка, модульные тесты, статический анализ и smoke; для API также контрактные проверки. Полный E2E исходно проводится на integration перед выпуском и не объявляется уже работающим обязательным тестом каждого MR.

После merge в main CI публикует образы с commit/digest. Согласованная ревизия переносится в integration; коммит запускает деплой на общую среду. Пара образов web/API фиксируется явно, между средами образы не пересобираются. У независимого изменения web может сохраниться прежний API. Совместимость контрактов и миграций проверяется отдельно.

Feature-среды `portal-mr-<номер>` в продуктовом Kubernetes — планируемое изменение по инфраструктурной карточке. До использования нужно проверить создание среды и данных, точность ревизии, блокировку конкурирующих деплоев, готовность и очистку. CI-runner развёртывает через Kubernetes API, затем выполняет проверки доступности приложения. Журнал деплоя связывает pipeline, среду и digest; запоздалый pipeline не должен подменить уже проверяемую ревизию.

После тестирования Анна принимает результат на integration. Обычные окна production — вторник и четверг 16:00–17:00 по Москве. Анна подтверждает состав, Вера или заместитель — результаты проверок, дежурный платформы — доставку и технический контроль. Только установка нужных образов и контрольные действия позволяют отметить «Выпущено». Возврат образов возможен при совместимой схеме БД; универсальный откат миграций не предполагается.

5. События для мониторинга

Интервал	Начало → конец в примере	Источники и ограничения
Очередь ревью	Готовый MR/SHA и доставленный запрос → начало содержательного ревью	GitLab и Mattermost; начало работы ещё нужно сделать наблюдаемым
Рассмотрение	Начало ревью → вердикт по SHA	Review/discussions/approval. Это длительность рассмотрения, не автоматически активные часы человека
Ожидание merge	Все одобрения и обязательные проверки актуальны → merge	GitLab и CI; draft и исправления сюда не входят

Интервал	Начало → конец в примере	Источники и ограничения
Очередь CI	Job без незавершённых зависимостей принята к исполнению → runner начал работу	GitLab CI. Ожидание предыдущей job относится к зависимости, не занятости runner
Среда	Требуется среда ревизии → доступна нужная ревизия и данные, health check успешен	CI и резервирование; до появления feature-сред учитывается общая integration
Отказ инфраструктуры	Ошибка блокирует шаг → доступность восстановлена и проверена	Логи CI, GitLab, registry и среды; не вся длительность общего инцидента приписывается каждой задаче

Начало очереди тестировщика определяется после готовности среды и передачи комплекта проверки, по ЖЦ задач. В план мониторинга включаются эти ожидания и очереди Джины. Единый ID ожидания связывает записи разных систем; повторный сбор не размножает интервал. Открытые ожидания, недостаток событий и размер наблюдаемой выборки раскрываются рядом со статистикой. Фактических измерений и подтверждений настройки в этом примере пока нет.

Карточка инфраструктуры

Карточка описывает инфраструктуру внедрения в продукт: инференс, ИИ-платформу, инструменты конвейера разработки и тестовые среды. В ней собраны размещение систем, необходимые сетевые связи и доступы людям, системам и ИИ-платформе. Отдельные реестры Git, аккаунтов и связности не требуются: существующие технические описания и результаты проверок подключаются ссылками.

Пример заполнения: личный кабинет Меридиан Сервис.

Шаблон

Скопируйте форму ниже в рабочий документ проекта. Требования и фактическую готовность заполняйте раздельно: выбранная система и выданный аккаунт ещё не означают, что необходимые операции работают.

Инфраструктура внедрения в продукт «…»

Паспорт проекта: …

Версия и дата актуальности: …

Для какого этапа описана конфигурация: Подготовка / PoC / Пилот. …

1. Конфигурация

Группа	Выбранное решение и область использования	Что переиспользуется / что нужно подготовить
Инференс	Поставщик, размещение, модели и назначения, API, квоты и ограничения, порядок получения сведений о расходах: …	…
ИИ-платформа	Продукт и версия, экземпляр, проект команды, где выполняются агенты и их инструменты: …	…
Конвейер разработки	Трекер и проект, Git и репозитории, CI/CD, анализ кода, артефакты/образы, чат, документация, логи и мониторинг: …	…
Тестовые среды	Среды и назначение, состав продукта и зависимостей, данные, E2E, управляемые моки, время и другие необходимые механики: …	…

Условия передачи данных внешним сервисам: какие сведения передаются, ограничения, хранение и использование поставщиком, основание допустимости. …

2. Размещение и ресурсы

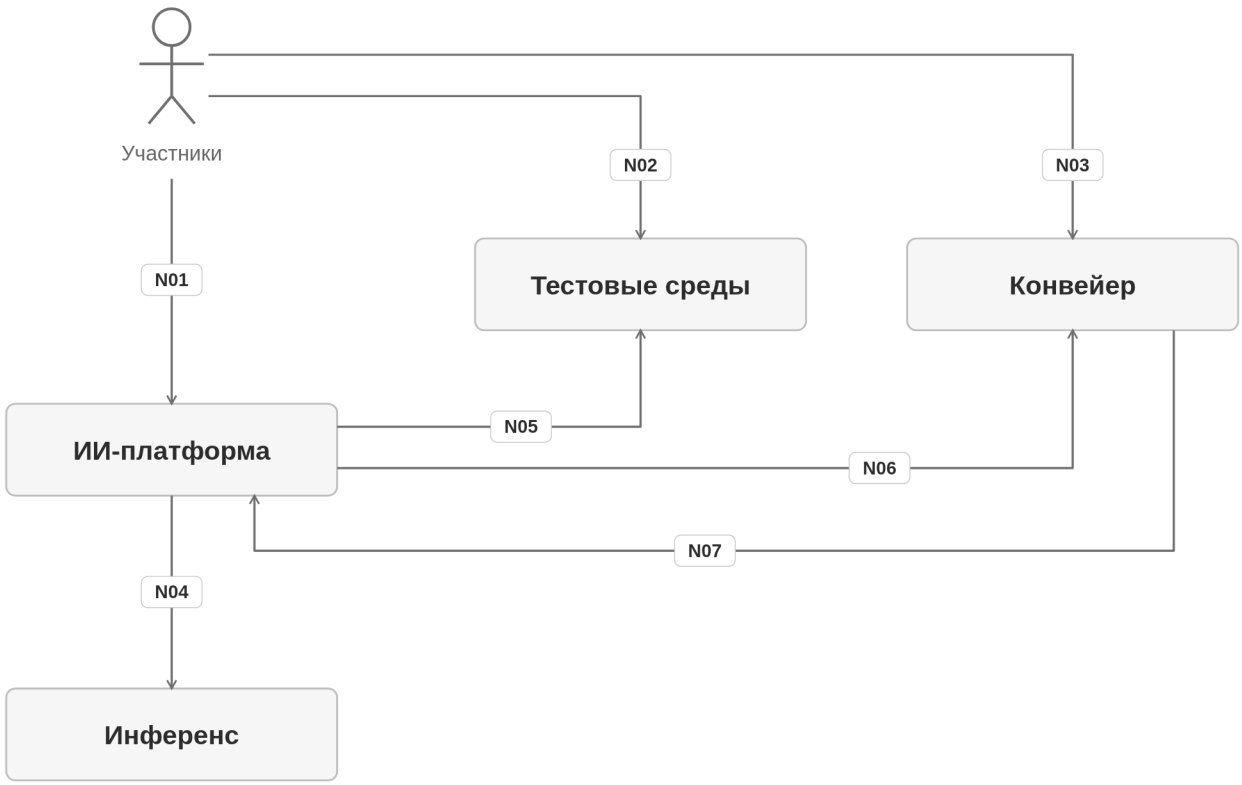
Под ресурсами понимаются процессор, оперативная память и место на диске, которые нужно выделить системам. Хранение — где сохраняются настройки, память агентов и рабочие материалы после перезапуска. Место выполнения агентов — где запускаются команды, сборки и тесты: например, во временных контейнерах Kubernetes предприятия. Эти сведения нужны для развёртывания, а не для описания людей или их рабочих мест.

Одна строка — один экземпляр системы или среды. Обозначения используются далее в связях и доступах.

Система / экземпляр	Площадка, сетевой контур и адрес	Требуемые ресурсы / квоты	Фактическое состояние и подтверждение
…	ЦОД/облако, сервер или кластер/namespace; URL/API: …	CPU, RAM, диск, параллельность либо квота сервиса; основание: …	Выделено/не выделено; версия, дата, ссылка на проверку: …

3. Сетевая связность

Типовая архитектурная схема



Стрелка — инициатор подключения → адресат. Коды N01–N07 соответствуют таблице связности.

Открыть схему крупно.

Схема показывает четыре группы: **ИИ-платформа, инференс, тестовые среды и конвейер**. ИИ-платформа включает работу агентов и их инструментов; конвейер — трекер, Git, CI, документацию, чат, хранилища и другие используемые системы. Инференс может быть внутренним или внешним. Внутреннее устройство групп, размещение и границы сетей раскрываются в проектной карточке по необходимости.

Стрелки показывают инициатора подключения и адресат. Ответы возвращаются по тем же соединениям; обратные вызовы из конвейера выделены отдельно. Подключения внутри конвейера, доступ CI к тестовым средам для развёртывания, health check и smoke и связи приложения с тестовыми зависимостями здесь подразумеваются. Их требования и проверки фиксируются в рабочей карточке, без обязательного добавления на верхнеуровневую схему.

Таблица связности

Замените группы конкретными системами и адресами. Если одному коду соответствуют разные адресаты, протоколы или права, разделите строку с уточнением кода, например N06.1 и N06.2. Неприменимые связи отметьте и уберите со схемы; дополнительные внесите в оба места.

Код	Откуда → куда	Назначение	Маршрут, протокол и ограничения	Проверка из фактического контура
N01	Участники → ИИ-платформа	Поручение задач, настройка и просмотр результатов	Адреса, способы входа и разрешённые действия: …	Личный вход и операции по роли; дата и подтверждение: …
N02	Участники → тестовые среды	Просмотр результата и проверки специалистами	Адреса сред, роли пользователей, доступ к данным и диагностике: …	Вход и проверка результата; дата и подтверждение: …
N03	Участники → конвейер	Задачи, код, MR, документация, сообщения и результаты проверок	Системы, проекты, адреса и личные аккаунты: …	Необходимые действия участников; дата и подтверждение: …
N04	ИИ-платформа → инференс	Вызовы моделей и получение ответов	Сервис, endpoint, протокол, ограничения передаваемых данных и квоты: …	Запрос/ответ выбранных моделей и сведения об использовании; дата и подтверждение: …
N05	ИИ-платформа / агенты → тестовые среды	Проверки поведения, E2E, получение логов и разрешённая диагностика	Адреса сред и средств диагностики, протоколы, тестовая область и служебные права: …	Прогон и получение диагностики; дата и подтверждение: …
N06	ИИ-платформа / агенты → конвейер	Чтение задач и контекста, работа с кодом/MR, сообщения, запуск проверок и чтение результатов	Системы, проекты, адреса и операции служебных аккаунтов: …	Задача с вложением, изменение кода/MR, обращение к человеку и результат проверки; дата и подтверждение: …

Код	Откуда → куда	Назначение	Маршрут, протокол и ограничения	Проверка из фактического контура
N07	Конвейер → ИИ-платформа	Входящие события и предусмотренные подключения средств конвейера: webhook, сообщения, установка компонентов через CI по применимости	Конкретный источник и адресат каждого подключения, протокол, права и проверка подлинности: … / не применяется	Доставка/обработка события или выполнение предусмотренного задания; дата и подтверждение: …

Другие необходимые подключения: развёртывание и прямые проверки сред из CI, зависимости тестовых приложений, DNS/TLS, загрузка пакетов и образов, хранение данных и резервное копирование. Адресаты, маршруты, полномочия и подтверждения по применимости: …

4. Учётные записи и доступы

Значения паролей, ключей и токенов не записываются. Указываются способ выдачи и защищённое место хранения, срок действия и порядок продления/отзыва.

Людям

Участник и роль	Ресурс и разрешённые действия	Учётная запись, способ входа и срок	Проверка участником
…	Проект/репозиторий/среда, конкретные операции: …	…	Дата, выполненные действия, результат, подтверждение: …

Системам

Система / компонент	Ресурс и разрешённые действия	Служебная учётная запись и способ подключения	Проверка операций
CI-runner, приложение или другой компонент: …	…	Срок и порядок работы с секретом: …	Из какого контура, дата, результат, подтверждение: …

ИИ-платформе

Компонент платформы / среда агента	Ресурс и разрешённые действия	Учётная запись и неинтерактивное подключение	Проверка операций
…	Инференс, трекер, Git, CI, среды, чат, документация; конкретная область: …	Срок и порядок работы с секретом: …	Из какого контура, дата, результат, подтверждение: …

Границы полномочий: защищённые ветки, слияние, выпуск в production, изменение прав и другие запрещённые или требующие участия человека действия. …

5. Тестовые среды и технический прогон

Развёртывание и восстановление: как получить среду из выбранной ревизии, состав приложения и зависимостей, сброс данных и восстановление после неудачной пробы. …

Проверки поведения: как агент запускает E2E и получает результаты, логи, скриншоты и другие материалы для разбора. …

Управляемые условия: данные и роли пользователей, программно-управляемые моки внешних систем и побочных эффектов, виртуальное время, эмуляторы оборудования — по применимости выбранных сценариев. Неподдержанные сценарии и ограничения: …

Подтверждение технического прогона: ревизия → сборка → артефакт → развёртывание → проверка поведения → диагностика; дата и ссылки на результаты. …

6. Различия PoC и Пилота, изменения конфигурации

Компонент / условие	На PoC	На Пилоте	Что перенастроить и повторно проверить
…	Временный или целевой инструмент, данные и ограничения: …	…	…

История изменений: дата/период применения, изменение конфигурации, основание и подтверждение проверки. …

Связь с затратами: где сохранены параметры тарификации и сведения о расходах по периодам конфигураций; разовые работы отделены от дальнейшей эксплуатации. …

7. Согласование и готовность

Конфигурация и необходимые доступы согласованы: версия, участники решения, дата и ссылка. …

Системы, сети и служебные подключения проверены: итог и ссылки на подтверждения из разделов 2–5. …

Личные доступы проверены участниками: итог и ссылки из раздела 4. …

Незавершённые обязательные работы: ссылка на план / отсутствуют. …

Подтверждение готовности к соответствующему этапу: ссылка на чек-лист и результаты проверки. …

Как заполнять

На Подготовке выбирают конфигурацию и описывают необходимые ресурсы, связи и операции. После настройки в тех же строках сохраняют фактические проверки. Перед Пилотом уточняют отличия от PoC и повторяют проверки изменённых подключений; во время Пилота фиксируют

существенные изменения и периоды их применения. Карточка остаётся одним документом на протяжении внедрения.

- **Конфигурация и размещение.** Для on-prem укажите площадку и контур предприятия, место работы агентов и инструментов, ресурсы, хранение рабочих данных и восстановление. Для внешнего инференса — поставщика, API, модели, квоты, стоимость и допустимость передачи данных. On-prem платформа с внешним инференсом всё равно передаёт поставщику содержимое запросов к модели; размещение платформы само по себе не означает, что код остаётся внутри предприятия.
- **Сеть.** Опишите действительные источники соединений. Проверка с ноутбука не подтверждает доступ из среды агента, платформы или CI-runner. Обратные связи указывайте только для используемой интеграции; отдельно проверяйте хранилища вложений трекера и документации.
- **Доступы.** Назовите необходимые операции, а не только систему. Например: читать задачи и вложения, создавать ветки и MR, запускать проверки, получать их логи. Не предполагается, что агенту доступно всё, что доступно человеку. Интерактивная учётная запись не заменяет служебное подключение.
- **Среды.** Агентам требуется воспроизводимый способ собрать и проверить продукт, включая E2E в среде, достаточно близкой к production для выбранных задач. Укажите конкретные механики управления данными, зависимостями и временем; существование dev-стенда не доказывает их готовность. Реальное оборудование заменяется управляемым эмулятором, если оно входит в проверяемый сценарий.
- **Переход к Пилоту.** Временные инструменты PoC допустимы. Запишите, что переносится и что сохраняется; взаимодействия с людьми должны представлять будущий процесс, а в целевом контуре нужно проверить затронутые подключения и рабочий маршрут. Неизменённые компоненты не требуют повторного развёртывания ради новой фазы.
- **Подтверждения.** Сеть, аутентификация и прикладные операции проверяются отдельно. Ссылки на CI и журналы предпочтительнее повторного ручного описания логов. Не ставьте отметку готовности по плану установки или факту выдачи аккаунта.

Подготовку карточки организует ИИ-инженер с участием DevOps и специалистов продукта. При on-prem DevOps команды внедрения может устанавливать платформу, а специалист предприятия — предоставлять ресурсы и подключения. Конкретные работы и один ответственный за каждую указаны в плане; карточка не создаёт новых ролей. Служебные подключения подтверждает ИИ-инженер, личные проверки участников организует РП.

Инфраструктура внедрения в личный кабинет Меридиан Сервис

Шаблон и заполнение · Паспорт проекта.

Версия и дата актуальности: плановая конфигурация первого внедрения. Исходное состояние предприятия описывается на 4 мая 2026 года; приведённые ниже настройки предстоит выполнить на Подготовке и PoC. Успешные проверки не заявляются заранее.

Этапы: Подготовка — 1–21 июня; PoC — 22 июня–26 июля; Пилот — 27 июля–18 октября 2026 года.

Размещение: Джина в отдельном Kubernetes-кластере предприятия. Её кластер создаётся в проекте; продуктовые тестовые среды размещаются в другом, уже существующем кластере. Инференс — внешний сервис компании «Практика ИИ», которая ведёт внедрение. С PoC используются действующие GitLab, Jira, Mattermost и Confluence. Временные копии этих систем не создаются.

Домены `meridian.example` и `praktika-ai.example`, имена namespace и аккаунтов ниже — условные обозначения учебного примера, а не действующие адреса или уже созданные ресурсы.

1. Конфигурация

Группа	Выбранное решение и область использования	Что переиспользуется / что нужно подготовить
Инференс	Сервис «Практики ИИ», модели **GLM 5.3** и **GLM 5.3 flash**. GLM 5.3 — основная модель для решения задач и сложных разборов; flash — для коротких вспомогательных операций, пригодность проверяется на PoC	Выделенный проект `meridian-portal`, API `https://llm.praktika-ai.example`, отдельный ключ и квота. Распределение операций между моделями настраивает Никита Логинов
ИИ-платформа	Джина в отдельном Kubernetes-кластере «Меридиан Сервис», который создаётся для платформы и рабочих контейнеров агентов. Общий экземпляр предприятия — `gina-meridian`; компоненты платформы — namespace `gina-meridian`, рабочие контейнеры агентов — `gina-meridian-workspaces`. Первый подключаемый продукт — кабинет; экземпляр предназначен и для следующих продуктов программы	Тимур предоставляет узлы и сетевые подключения; Арсений Дроздов разворачивает отдельный кластер, Джину и исполнение агентов. Джина программно управляет своим кластером; управление собственным кластером отделено от ограниченного доступа к тестовым средам продуктового кластера для логов и диагностики. Конкретная версия и размеры выделяемых мощностей уточняются перед установкой
Конвейер разработки	Действующие GitLab и CI, Jira `PORTAL`, Confluence, Mattermost, хранилища образов/артефактов, логи и мониторинг предприятия	Подключаются `customer-portal-web` и `customer-portal-api`, продуктовые задачи, документация и проектный канал. Новые Jenkins или SonarQube не вводятся

Группа	Выбранное решение и область использования	Что переиспользуется / что нужно подготовить
Тестовые среды	Существующая общая среда `integration` и новые feature-среды в Kubernetes	Сейчас коммит в ветку `integration` запускает деплой на общую среду. Feature-сред нет. В проекте создаются изолированные среды для веток/MR с собственными данными, сборкой, E2E и диагностикой

Передача данных «Практике ИИ»: разрешены только запросы агентов к LLM и получение ответов. Контекст запроса может включать необходимые для задачи постановки, фрагменты кода, документации и диагностики с соблюдением корпоративных ограничений на секреты и персональные данные. Выгрузка репозиториев, баз, журналов, резервных копий или телеметрии в другие сервисы «Практики ИИ» для хранения, сопровождения либо аналитики не предусматривается.

Доступ специалистов внедрения к системам предприятия организуется внутри его контура через персональный VPN; материалы остаются в системах «Меридиана». Это не отдельный канал экспорта данных поставщику. Условия хранения запросов и ответов у сервиса инференса ещё предстоит зафиксировать; вторичное использование переданных материалов не считается разрешённым самим фактом допуска LLM.

2. Размещение и ресурсы

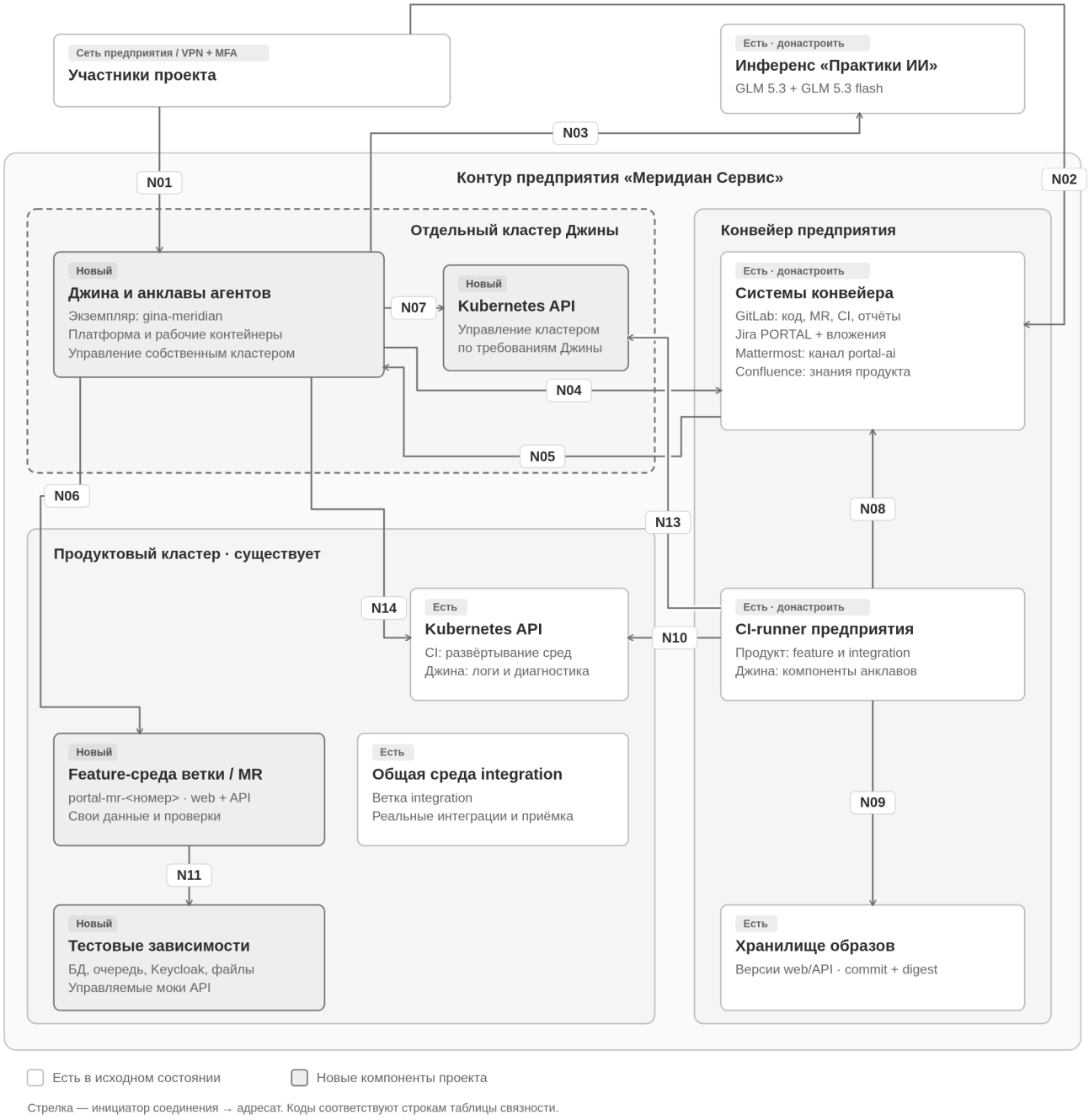
Система / экземпляр	Площадка, сетевой контур и адрес	Что нужно выделить или сохранить	Состояние
Джина	Новый отдельный кластер Джины, namespace `gina-meridian`; `https://gina.meridian.example`, доступ из корпоративной сети/VPN	Процессор, память и диски по требованиям выбранной версии. Настройки, история работы и память агентов сохраняются на постоянных дисках; резервные копии — внутри предприятия	Требуется установка
Рабочие контейнеры агентов	Отдельный кластер Джины, namespace `gina-meridian-workspaces`	Здесь агенты получают код и запускают инструменты, сборки и проверки. Рабочая копия кода временная; изменения сохраняются в Git, значимые результаты — в системах проекта. Лимиты параллельных работ определяются на PoC	Требуется настройка
Инференс	Внешний сервис «Практики ИИ», `https://llm.praktika-ai.example`	Доступ к GLM 5.3 и GLM 5.3 flash, квота проекта, сведения об использовании моделей и токенах	Требуется подключение

Система / экземпляр	Площадка, сетевой контур и адрес	Что нужно выделить или сохранить	Состояние
GitLab, CI и хранилища	Контур разработки: `git.meridian.example`, `registry.meridian.example`; артефакты CI через GitLab	Переиспользуются действующие системы и runner; для feature-сред добавляются задания CI	Действующие системы; новые сценарии нужно настроить
Jira и вложения	`jira.meridian.example`, отдельное хранилище `attachments.meridian.example`	Проект `PORTAL`, описания, комментарии и вложения; доступ проверяется к обоим адресам	Нужно подключить Джину
Документация и чат	`docs.meridian.example`, `chat.meridian.example`	Пространство кабинета и канал `portal-ai`; права на содержимое и вложения	Требуется настройка интеграций
Общая среда кабинета	Существующий продуктовый кластер, namespace `portal-integration`; `https://portal-integration.meridian.example`	Одна постоянная тестовая среда, связанная с веткой `integration`; используется для общей проверки и приёмки	Существует в исходном состоянии
Feature-среды	Существующий продуктовый кластер, новые namespace вида `portal-mr-<номер>`; `https://portal-mr-<номер>.meridian.example`	Web/API, отдельные тестовые данные и зависимости; квоты и удаление после завершения работы	Создаются в рамках проекта

Численные размеры CPU, RAM, дисков и квоты LLM не выдумываются: Арсений фиксирует требования к узлам отдельного кластера Джины и квотам feature-сред в продуктовом кластере; Тимур Агеев обеспечивает их выделение. Первые прогоны дают основание уточнить размеры и параллельность. Джина, агенты и feature-среды работают в инфраструктуре предприятия; снаружи выполняются только запросы к моделям.

3. Сетевая связность

Архитектурная схема



Открыть схему крупно.

Стрелки показывают **инициатора соединения и его адресат**, а не движение задачи между людьми и агентами. Ответы возвращаются по тем же соединениям; отдельная обратная стрелка нужна, если другая система сама инициирует вызов, как webhook N05. Пересечение линий без точки не означает их соединения. Коды соответствуют строкам таблицы ниже.

На схеме объединены одинаковые по назначению подключения: N04 раскрывает доступ к GitLab, Jira и её хранилищу вложений, Mattermost и Confluence. Это отдельные проверяемые адресаты, хотя показаны одним блоком. Зависимости общей integration не детализируются на верхнем уровне; их состав раскрыт в разделе тестовых сред.

Пунктиром выделен новый отдельный Kubernetes-кластер Джины; продуктовые среды размещаются в существующем продуктовом кластере. N07 относится к API кластера Джины, N10 — к API продуктового кластера. N13 показывает установку CI-раннером компонентов анклавов в кластер Джины. N14 — отдельный доступ Джины к тестовым namespace продуктового кластера для логов и диагностических подов. Реквизиты каждого кластера и назначения разделены. Нейтральные блоки уже существуют; бейдж «Есть · донастроить» означает необходимые подключения или изменения. Зелёные блоки с бейджем «Новый» создаются в проекте.

Схема отражает основные прикладные и управляющие соединения. CI-runner обращается к Kubernetes API, чтобы развернуть компоненты и среды. Kubernetes API не инициирует обращения к приложениям для health check или smoke. Такие проверки выполняет CI-runner, обращаясь непосредственно к тестовым средам; этот доступ подразумевается и отдельными стрелками не показан. В таблице он отмечен примечанием без кода. Внутреннее устройство Джины и служебные соединения Kubernetes здесь не детализируются.

Таблица связности

Перечислены проектируемые подключения. Правила реализуются в корпоративной сети и Kubernetes; их выполнение подтверждается техническими проверками.

Код	Откуда → куда	Назначение	Маршрут и ограничения	Проверка
N01	Участники проекта → Джина	Работа в UI, настройка и управление заданиями	Корпоративная сеть; специалисты «Практики ИИ» — персональный VPN с MFA. HTTPS/443	Личный вход и операции по роли; подтверждения собирает Марина
N02	Участники проекта → системы конвейера	Задачи, документация, чат, MR и результаты CI	Корпоративная сеть/VPN, HTTPS/443; персональные аккаунты и проектная область доступа	Вход, чтение материалов и действия каждого участника
N03	Компонент вызова моделей Джины → внешний инференс	Запросы GLM 5.3 / GLM 5.3 flash и получение ответов	Исходящий HTTPS/443 через корпоративный proxy; только llm.praktika-ai.example. Выход не открывает другие сервисы поставщика	Запрос и ответ обеих моделей, учёт использования
N04	Джина / среда агента → системы конвейера	Код, задачи и вложения, MR и комментарии, обращения в чат, чтение документации; запуск разрешённого CI и чтение результатов	HTTPS/443 к GitLab, Jira, отдельному хранилищу вложений, Mattermost и Confluence; ограничения служебных аккаунтов раздела 4	Задача с вложением → ветка/MR → pipeline и отчёт; вопрос → ответ человека → чтение ответа агентом

Код	Откуда → куда	Назначение	Маршрут и ограничения	Проверка
N05	GitLab / Mattermost → Джина	События MR, pipeline и сообщения проектного канала	Внутренний HTTPS/443 к endpoint интеграции Джины; проверка секрета webhook	Доставка события и обработка без повторного запуска той же работы
N06	Среда агента → своя feature-среда	Браузерный E2E и проверка поведения приложения	HTTPS/443 к URL выделенной среды. Браузер выполняется внутри предприятия; логи и отчёты доступны через N04	Запуск E2E, получение отчёта, скриншотов и диагностики
N07	Управляющий компонент Джины → API отдельного кластера Джины	Создание, завершение и очистка рабочих контейнеров агентов	TLS/6443 внутри отдельного кластера Джины. Права управления namespace, рабочими контейнерами и другими необходимыми объектами определяются требованиями устанавливаемой версии. Для тестовых сред продуктового кластера используется отдельная учётная запись N14	Рабочая сессия создаётся и очищается в кластере Джины; права её управления отделены от диагностического доступа N14
N08	CI-runner → GitLab	Получение заданий, кода и публикация результатов проверки	HTTPS/443 к действующему GitLab; служебные реквизиты CI	Получение кода выбранной ревизии и сохранение результатов pipeline
N09	CI-runner → хранилище образов	Публикация собранных образов web/API	HTTPS/443 к registry.meridian.example; доступ только к разрешённым проектам	Публикация образов, фиксация commit и digest
N10	CI-runner → API продуктового кластера / механизм выделения сред	Создание feature-среды и развёртывание образов; доставка в общую integration по действующему процессу	TLS/6443 к API продуктового кластера. Выделение namespace — через ограниченный механизм платформы. Доступ задания развёртывания — только к выданной среде, без cluster-admin и production	Развёртывание конкретного digest, обновление и удаление feature-среды; отдельно — разрешённый деплой integration

Код	Откуда → куда	Назначение	Маршрут и ограничения	Проверка
N11	Приложение feature-среды → её тестовые зависимости	Данные, очередь, файлы, идентификация и моки API планирования/parts-integration	PostgreSQL — 5432; RabbitMQ — 5671/TLS; тестовый Keycloak, файлы и API моков — HTTPS/443. Свои данные и тестовые реквизиты	Прикладной сценарий, сброс состояния, ошибки и повторы моков; изоляция данных разных MR
N13	CI-runner → API кластера Джины	Развёртывание и обновление компонентов анклавов в кластере Джины	TLS/6443; отдельные реквизиты CI для кластера Джины и разрешённых ресурсов анклавов, без использования ключа продуктового кластера	Компоненты анклава установлены из выбранной версии; готовность и обновление проверены
N14	Джина / среда агента → API продуктового кластера	Чтение логов и событий тестовых сред, создание и удаление временных диагностических подов	TLS/6443; отдельный service account для разрешённых тестовых namespace. Права на логи, события и диагностические поды; без cluster-admin и production. Диагностические поды ограничены квотами и правилами тестового контура	Агент читает логи своей среды, запускает и удаляет диагностический под; запросы за пределами разрешённых тестовых namespace отклоняются

Доступ CI к тестовым средам (без стрелки и кода). CI-runner обращается к URL развёрнутых тестовых сред по HTTPS/443 для health check и smoke. Проверяется успешный вызов и ожидаемый результат после доставки. При настройке сети этот маршрут предусматривается отдельно от N10 и N13; успешный вызов Kubernetes API не подтверждает доступность приложения.

DNS, внутренние сертификаты, загрузка базовых образов и пакетов, получение образов узлами Kubernetes и резервное копирование используются через действующие сервисы предприятия; при подготовке сетевых правил их служебные зависимости уточняют отдельно. Они не показаны стрелками N01–N11, N13, N14. Прямой выход рабочих контейнеров в интернет по умолчанию не открывается. Feature-среды не подключаются к production и не используют его БД или секреты. Выход к LLM настраивается у компонента Джины, выполняющего вызовы моделей; название компонента и маршрут сверяются с устанавливаемой версией.

4. Учётные записи и доступы

Людям

Каждый работает под собственной учётной записью; приведённые имена — план выдачи, а не уже созданные аккаунты. Доступ специалистов внедрения действует на срок проекта и пересматривается при изменении участия.

Участник и роль	Разрешённые действия	Способ доступа	Подтверждение
Никита Логинов, ИИ-инженер «Практики ИИ»	Настраивать проект Джины, инструкции, навыки и интеграции; читать код и задачи, вести MR и прогоны, работать с feature-средами и диагностикой	Персональный VPN/MFA и аккаунты предприятия, доступ в пределах кабинета	Проверяется после выдачи
Арсений Дроздов, DevOps «Практики ИИ»	Устанавливать и обновлять Джину, готовить рабочие контейнеры и feature-среды, настраивать CI и подключения	VPN/MFA; права установки и сопровождения отдельного кластера Джины. В продуктовом кластере — делегированные права тестовых сред; общекластерные изменения там выполняет Тимур	Проверяется после выдачи
Денис Пахомов, ИИ-инженер продукта	Настройки проекта, навыки, прогоны и диагностика по мере обучения; изменения совместно проверяет Никита	Персональная учётная запись предприятия	Проверяется на Подготовке и при расширении прав
Тимур Агеев, DevOps предприятия	Выделять ресурсы, настраивать корпоративную связность, DNS/TLS, служебные аккаунты и права в рамках полномочий предприятия	Действующий административный доступ	Дополнительные подключения проверяются отдельно
Илья, Сергей, Игорь, Елена, Вера и Ксения — специалисты кабинета	Действия своей роли: постановки, MR, тесты, приёмка; доступ к Джине и диагностике проекта	Персональные аккаунты предприятия	На PoC — обучение и демонстрация, на Пилоте — непосредственная работа
Марина Фомина, Кира Воронцова и Анна Лебедева — РП и владелец продукта	Задачи, коммуникации, результаты, отчётность; решения по процессу и продукту в своей роли	Персональные аккаунты; Кира — через VPN/MFA	Личные доступы и необходимые операции проверяются до использования

Проверки личного доступа не заменяются демонстрацией: каждый участник входит и выполняет необходимые для своей роли операции. РП собирает подтверждения. Обучение не требует самостоятельного ведения задач с агентами на PoC.

Системам

Учётная запись	Ресурс и операции	Ограничения и хранение	Проверка
`svc-portal-ci`	Получить код, опубликовать образ, развернуть/обновить/удалить feature-среду, выполнить E2E	CI запускает подготовленный механизм создания сред, разрешающий только namespace `portal-mr-*`, утверждённые квоты и тестовый контур. Cluster-admin не выдаётся. Секреты — в защищённых настройках CI	Полный цикл среды и запрет доступа к production
`svc-gina-enclaves-ci`	Развёртывать и обновлять компоненты анклавов Джины	Отдельные реквизиты кластера Джины в защищённых настройках CI; права на предусмотренные установкой ресурсы анклавов, без подмены реквизитами продуктового кластера	Установка выбранной версии и проверка готовности компонентов
`portal-feature-app`	Свои тестовые БД, очередь, файлы, тестовая идентификация и моки	Отдельные реквизиты среды в Kubernetes Secrets, без production-доступов	Проверки приложения и изоляции

Создание namespace требует отдельного механизма платформы: одного соглашения об имени недостаточно. Тимур предоставляет ограниченный механизм выделения сред, Арсений подключает его к CI. Доступ обычного задания развёртывания ограничивается выданной ему средой.

ИИ-платформе

Учётная запись	Ресурс и операции	Ограничения и хранение	Проверка
Ключ проекта `meridian-portal`	Обе модели сервиса «Практики ИИ», чтение доступной статистики использования	Отдельная квота проекта; ключ хранится в Kubernetes Secret Джины и не выдаётся приложению или обычным рабочим контейнерам	Вызовы моделей и получение использования
`svc-gina-portal-git`	Читать оба репозитория, отправлять рабочие ветки, создавать MR и комментарии, запускать разрешённые проверки и читать отчёты	Нет прямой записи в защищённые `main` и `integration`, обхода проверок, самостоятельного слияния и выпуска в production	Рабочая ветка/MR/pipeline; запрещённые операции отклонены
`svc-gina-portal-jira`	Читать задачи `PORTAL`, комментарии и вложения; оставлять сообщения, менять назначение на агентское и статусы по согласованному маршруту	Без администрирования, удаления задач, изменения проектов и самостоятельной фиксации продуктовой приёмки	Чтение вложения и разрешённые переходы

Учётная запись	Ресурс и операции	Ограничения и хранение	Проверка
`gina-portal-bot`	Читать и писать в `portal-ai`, обрабатывать ответы и упоминания	Только проектный канал; без чтения чужих личных переписок и прочих каналов	Цикл вопрос → ответ → продолжение работы
`svc-gina-portal-docs`	Читать пространство кабинета и его вложения	Без редактирования исходной документации; инструкции и навыки сохраняются отдельно в проектном Git	Чтение страницы и файла
`gina-workspace-controller`	Управлять рабочими контейнерами и необходимыми объектами отдельного кластера Джины	Служебная учётная запись отдельного кластера. Объём прав задаётся требованиями устанавливаемой версии. Для тестовых сред другого кластера используется отдельная учётная запись N14	Создание и очистка рабочей сессии; разделение реквизитов двух кластеров
`svc-gina-portal-diagnostics`	Читать логи и события разрешённых тестовых namespace, создавать и удалять временные диагностические поды	Учётная запись продуктового кластера с областью `portal-integration` и выданных проекту feature-сред. Поды ограничены правилами и квотами тестового контура; без cluster-admin и production	Чтение логов, создание и очистка пода; отказ в чужой области

Служебные аккаунты не заменяются персональными. Срок действия, продление и отзыв фиксируются при выдаче; секреты хранятся внутри предприятия, не в карточке и не в Git. Для мониторинга используются те же доступные проектные источники и отдельные права чтения при необходимости; массовые выгрузки во внешние сервисы не вводятся.

5. Тестовые среды и технический прогон

Исходное состояние. У кабинета одна постоянная тестовая среда `integration`, деплой в неё запускается коммитом в одноимённую ветку. На ней проходят интеграционные проверки, регрессия и приёмка; отдельной stage или feature-сред нет. Общие данные и асинхронные обновления делают шесть из 22 E2E нестабильными. Сброс общей среды требует согласования с пользователями стенда.

Что организуется в проекте. Арсений готовит CI для feature-сред в существующем продуктовом Kubernetes-кластере, Никита — команды, которыми агент получает среду, запускает проверки и читает результат. На ветку/MR выделяется изолированная среда с web/API и собственными синтетическими данными. Повторный запуск для того же MR обновляет её; по закрытию MR среда удаляется. До завершения работы можно повторять цикл исправления и проверки без перезаписи общей `integration`.

Состав. Feature-среда содержит PostgreSQL, RabbitMQ, тестовое файловое хранилище и программно-управляемые моки API планирования и `parts-integration`. Для входа используется отдельный тестовый realm Keycloak без реальных пользователей. Данные создаются из

версионируемого набора; предусмотрены сброс, повторное создание и отдельные роли заказчиков для проверок доступа. Общие зависимости не должны связывать состояния разных feature-сред.

Управляемые сценарии. Моки позволяют задавать ответы, задержки, ошибки и повторы, читать журнал обращений. Для проверок асинхронных статусов предусмотрена программная публикация событий в тестовую очередь и ожидание конкретного результата. Для сценариев, зависящих от времени, добавляется тестовый механизм задания времени и запуска регламентных операций, доступный только в тестовой конфигурации. Эти механики и наборы данных создаёт «Практика ИИ» с предметной проверкой Веры и Сергея. Физическое оборудование в сценариях кабинета не участвует.

Проверки. Никита выбирает обязательные E2E для области агентов и разбирает шесть нестабильных сценариев: устраняет влияние общих данных и ненадёжных ожиданий либо фиксирует конкретное оставшееся ограничение. Результат включает отчёт, скриншоты и логи, привязанные к MR и версии образов. Перезапуск до первого зелёного результата без разбора причины не считается проверкой.

Реальные интеграции. Feature-среды не подменяют проверку контрактов и взаимодействия с настоящими системами. Общая `integration` сохраняется для этих проверок и приёмки. В её составе и календаре фиксируются совместимые версии зависимостей и окна использования; сценарий, требующий отсутствующей интеграции, не объявляется проверенным по одному моку.

Технический прогон до PoC. Существующая ревизия → сборка → образы → feature-среда → базовый E2E → получение диагностики и очистка среды. Детальные сценарии задач, навыки и дополнительные механики уточняются на PoC. Ссылки на успешные прогоны заполняются по выполнению работ.

6. Различия PoC и Пилота, изменения конфигурации

Компонент / условие	На PoC	На Пилоте	Что подтверждается перед Пилотом
Git, Jira, документация и чат	Целевые системы предприятия	Те же системы	Актуальные права, интеграции и рабочий маршрут; перенос из временных инструментов не требуется
Джина и инференс	Отдельный on-prem кластер Джины; GLM 5.3 / GLM 5.3 flash. Первоначальная настройка и работа — у Никиты	Та же конфигурация с уточнёнными по прогонам лимитами и версиями	Готовность выбранной версии, квот, диагностики и сопровождения
Среды	Создание и отработка feature-сред, настройки данных, моков и E2E	Использование в обычном потоке задач; общая `integration` сохраняется	Воспроизводимость развёртывания, сброса и проверок для согласованной области
Участие людей	Никита ведёт взаимодействие с агентами; предметные специалисты подключаются через него, проходят обучение и демонстрацию	Команда кабинета работает с агентами непосредственно; «Практика ИИ» помогает и сопровождает	Различия участия доведены до команды; материалы и доступы готовы

Сопровождение и распределение работ

Работа	Один ответственный	Участие остальных
Выделить ресурсы Kubernetes, корпоративные маршруты, DNS/TLS и исходные полномочия	Тимур Агеев, «Меридиан»	Арсений готовит требования и конфигурацию. Решения в инфраструктуре предприятия остаются у уполномоченных сотрудников
Развернуть и сопровождать отдельный кластер Джины, платформу, контейнеры агентов и CI feature-сред	Арсений Дроздов, «Практика ИИ»	Тимур обеспечивает разрешённые действия в общих системах. Никита проверяет рабочий маршрут
Подключить модели, навыки, контекст, коммуникации и диагностику; вести PoC	Никита Логинов, «Практика ИИ»	Денис учится на выполняемых работах. Илья, Сергей и Вера подтверждают предметные решения
Подготовить управляемые тестовые механики, данные и сценарии агентов	Никита Логинов, «Практика ИИ»	Арсений готовит среду и CI; Вера определяет нужные сценарии и проверяет результат; Сергей — изменения приложения
Организовать участие специалистов кабинета и проверить личные доступы	Марина Фомина, «Меридиан»	Кира координирует выделенную команду; каждый участник выполняет личные проверки
Устранить сбой внешнего инференса и восстановить предоставляемый сервис	Арсений Дроздов, «Практика ИИ»	Никита фиксирует влияние на задачи и проверяет восстановление. Внутренние обращения к службе сервиса организует «Практика ИИ»

На PoC «Практика ИИ» выполняет основную настройку, диагностику, доработки и сопровождение. Предприятие обеспечивает ресурсы и полномочия, доступность своих специалистов и продуктовые решения. На Пилоте Денис постепенно принимает освоенные работы; сложные случаи и сопровождение установки остаются у «Практики ИИ», пока передача конкретных работ не подтверждена. Завершение PoC само по себе не означает окончания сопровождения поставщиком.

История изменений: сохраняются версия платформы, инструкции развёртывания, параметры сред, моделей и лимитов, дата/период применения, причина и ссылка на проверку. Фактических изменений на исходном срезе нет.

Учёт ресурсов: отчёты об использовании моделей, квот и вычислительных ресурсов сохраняются внутри предприятия с привязкой к периодам конфигураций. Денежные значения в этом примере пока не заполняются.

7. Согласование и готовность

Карточка задаёт плановое устройство инфраструктуры и назначений. Имена аккаунтов и адреса показывают, что требуется настроить. Они не означают, что подключение уже выполнено или корпоративное согласование получено.

На Подготовке должны появиться ссылки на решения по конфигурации и доступам, проверки Джины и обеих моделей, Git/Jira с вложениями, чата и документации, feature-среды и личных входов. Системные проверки подтверждает Никита, личные доступы организует Марина. Перед Пилотом проверяется окончательный рабочий маршрут в тех же целевых системах.

До появления подтверждений пункты инфраструктурной готовности в гейтах не отмечаются выполненными.

Источники: предприятие, кабинет, сценарий внедрения и участники.

План мониторинга внедрения в продукт

План связывает цели внедрения с выбранными метриками, исходной базой, целевыми значениями, источниками данных и порядком наблюдения. Он нужен, чтобы команда одинаково понимала, какое изменение проверяется, по каким сведениям рассчитываются показатели и когда можно сделать вывод о Пилоте.

Документ заполняют на PoC до готовности к Пилоту. На Подготовке в PREP-02 формулируют только содержательную цель в паспорте; на PoC выбирают вычислимые метрики, текущие значения и предварительные ориентиры, на Пилоте уточняют расчёты и при необходимости целевые значения. Полный порядок — в гайде процесса. Цели и сроки берутся из паспорта проекта, определения и расчёты — из метрик команды. Это форма для продукта; сводный мониторинг программы предприятия здесь не описывается.

Пример: личный кабинет Меридиан Сервис на раннем PoC.

Шаблон

Скопируйте форму ниже в документ проекта. Неизвестные значения обозначайте явно: что нужно выяснить и какой работой PoC это будет проверено. На Подготовке заполнение плана не требуется. На раннем PoC можно заполнить только состав метрик, их смысл, границы и предполагаемые источники, оставив значения и подтверждения незаполненными. Затем выбранный набор, источники и предварительные ориентиры согласуются, сбор и расчёты проверяются к Пилоту.

План мониторинга продукта «…»

Паспорт проекта и цели внедрения: …

Версия плана, дата применения и ссылка на решение: …

1. Область и условия наблюдения

Поток работ: продукт, части системы, типы задач и исключения. …

Процессы и точки измерения: ссылки на жизненный цикл задач, Git/MR/проверки и предусмотренное участие людей; используемые события начала и завершения. …

Период наблюдения: даты из паспорта; календарь когорт, отсечки и дополнительное время наблюдения, если применяются; часовой пояс. …

Условия сопоставимости: сравниваемые потоки и состав работ, способ проверки качества, режим HITL/HOTL, версии агентов, конфигурации сред и инференса. …

Ограничения выводов: доступный объём наблюдений, неполные данные и известные различия. …

2. Выбранные метрики

Повторите блок для каждого показателя. Название должно обозначать число, долю, длительность, трудозатраты или стоимость. Рабочие значения за дни и недели сохраняются в сводках; здесь фиксируются правила и согласованные ориентиры.

M-… · Название метрики

- **Назначение:** результат / процесс / баланс.
- **Связь с целью или управленческим решением:** ссылка на цель паспорта; какой прогресс, препятствие или нежелательное последствие показывает метрика. …
- **Определение и единица:** ссылка на конкретную метрику в методологии либо определение проектной метрики; для доли — состав числителя и знаменателя. …
- **Проектные границы:** какие задачи/работы входят, период, события начала/конца, календарное или рабочее время, правила исключений. …
- **Исходная база:** значение и период / предстоит определить; зарегистрированные или оценочные сведения, источник и ограничения сопоставимости. …
- **Предварительное / уточнённое целевое значение или допустимая граница:** … / наблюдаем без порога; статус ориентира, основание и версия, срок проверки из паспорта, ссылка на согласование. …
- **Источники и необходимые сведения:** системы, события/поля, связь с задачей, MR, запуском, версией и участником; промежуточные оценки, если используются. …
- **Расчёт и проверка:** ссылка на стандартное правило или проектное дополнение, место подготовленного набора данных, версия навыка/алгоритма и подтверждение пробного вычисления. …

Обязательный охват ожиданий

В набор обязательно включаются ожидания ревью MR, тестирования и других применимых HITL, а также внутренние ожидания агентов: очередь готовых задач, зависимости/последовательное выполнение, лимит параллелизма, тестовые среды и инфраструктурные отказы. Существенные очереди CI и инференса выделяются отдельно. Это показатели процесса для управления; численные целевые пороги не обязательны для каждой причины.

Для каждого применимого вида указываются медиана длительности, число наблюдений, открытые ожидания и их возраст, полнота данных. Неприменимость обосновывается; отсутствие источника не считается отсутствием ожиданий. Границы событий берутся из ЖЦ задач и Git-процесса, единицы и перекрытия — из правил расчёта. Проектное сопоставление причин и источников: … .

3. Первичные сведения и оценочные значения

Первичные источники: трекер, Git, чат, журналы агентов, CI и тестовых сред; доступные сведения, способ получения и связи между ними. …

Оценки: идеальные часы выбранного объёма, участие людей и другие оценочные величины; агент / команда, основания, различение факта и оценки, сохранение версий и причин уточнения. …

Подтверждение оценок РП: порядок проверки, отбора аномалий и согласования в составе сводки; ссылка на результат согласования. …

Наборы для расчёта: место сохранения, версии, покрытие задач/дат и возможность повторно вычислить метрики. …

Полнота: как видны пропуски, открытые задачи и ожидания, незавершённые проверки и отложенные исправления; какие результаты пока предварительные. …

Границы затрат: в человеческий труд входят проверки, помощь, HOTL и ручные доделки по выбранному потоку. Общие работы внедрения и развития, эксплуатация и расходы платформы/инференса учитываются отдельно по применимым правилам, без двойного счёта. Проектные уточнения: …

4. Сбор и представление результатов

Ежедневный сбор: раз в сутки за прошлый календарный день; часовой пояс и время …; автозапуск / просьба агенту. …

Пропуски и пересборка: запуск за заданные дни или диапазон, сохранение покрытия дат, устранение повторов без двойного учёта. …

Источники результатов: дневной журнал, недельная сводка, артефакты расчёта и итог Пилота; ссылки. …

Недельный разбор: день/время, участники, проверка полноты и оценок, место решений об изменениях. …

Итоговая проверка: дата/окно из паспорта, набор показателей и обязательных условий завершения Пилота, место сохранения итогов. …

5. Участники и техническое обеспечение

Организует мониторинг и проверку сводок: … — РП; при двух назначениях РП указать распределение проверки и согласования.

Настраивает сбор, расчёты и диагностику: … — ИИ-инженер; навык/алгоритм и версия …; необходимые подключения подтверждены … .

Подтверждают предметные сведения и ограничения: участники продукта по применимости. …

6. Открытые вопросы и изменения

Что уточнить на PoC: вопрос, связь с метрикой, работа/ответственный и срок проверки. …

Подтверждение готовности к Пилоту: согласованная версия набора метрик, границ, ориентиров и источников; результаты пробного сбора, оценки исходной базы и недельной сводки. …

Изменения: дата и причина, изменённые показатели/правила/пороги/источники, период применения, решение и влияние на сопоставимость. …

Как заполнять

План организует РП вместе с владельцем продукта и ИИ-инженером. Роль не обязывает создавать отдельную команду мониторинга: один человек может совмещать назначения, специалисты продукта подключаются к своим предметным вопросам.

1. От цели — к измеряемому изменению

Возьмите цель из паспорта и уточните, какое наблюдаемое изменение покажет её достижение. «Использовать агентов» описывает средство; «сократить срок поставки выбранного потока» позволяет выбрать показатель результата. Не переписывайте цель отдельно в нескольких документах — сохраните ссылку и область измерения.

В PREP-02 цель формулируется по сути, без формул и численных показателей. К измерению переходят на PoC: проверяют доступные метрики и текущие значения, затем выбирают набор на Пилот и предварительные целевые ориентиры. Неизвестное значение не превращайте в обещанный процент. На Пилоте основания и процедуру расчёта уточняют, особенно оценку объёма задач; при необходимости согласованно корректируют и ориентиры.

2. Выбрать метрики результата, процесса и баланса

Начните с типовых метрик команды. Для каждой цели выберите метрику результата, которая показывает прогресс. Затем добавьте показатели процесса, по которым можно найти препятствие и выбрать действие, и баланса, по которым видно ухудшение качества, нагрузки или расходов. Группы метрик — назначения измерений, а не отдельные типы целей.

Что проверяем	Результат	Процесс для управления	Баланс
Ускорение поставки	Медианное время поставки	Длительность выбранных этапов и ожидание проверки	Возвраты с тестирования, дефекты после приёмки
Снижение труда людей на поставленный объём	Оценка экономии трудозатрат	Автономность по трудозатратам, передачи задач людям	Часы проверяющих, ручные доделки и качество
Снятие дорогого ручного контроля при сохранении результата	Та же метрика результата, что соответствует цели внедрения	Труд и ожидание HITL на сопоставимых работах	Возвраты с тестирования, дефекты после приёмки, очередь проверок

Таблица помогает выбрать сочетание, но не назначает весь набор каждому проекту. Не добавляйте метрику только потому, что её легко получить. Для процессной метрики должно быть понятно, какое решение примет команда при изменении значения. Например, «надежность агентов» слишком расплывчато; доля задач с дефектом после приёмки имеет определённую выборку и смысл.

3. Если подходящей метрики в каталоге нет

Сформулируйте величину и единицу, состав учитываемых работ, период и необходимые события. Для доли явно задайте числитель и знаменатель, для длительности — начало и конец, для затрат — включённые работы и способ получения значения. Укажите, какие записи позволяют проверить результат и в каких случаях вывод будет недостоверен. Проектное определение и дополнения к расчёту сохраняйте в блоке метрики, не выдавая новую величину за стандартную.

Для стандартной метрики достаточно ссылки на расчёт в `metrics.md` и проектных границ; формулу и справочник определений повторять не нужно.

4. Определить сравнение и целевые значения

Укажите исходную базу, её период и происхождение. Измеренное значение, экспертная оценка и предположение — разные основания. Для сравнения выберите сопоставимые работы и условия; не подменяйте эффект различием сложности задач или сокращением проверок. Результаты PoC не становятся автоматически исходной базой Пилота: могут отличаться участие инженера, инфраструктура и состав задач.

Для метрик результата задайте ожидаемое изменение и срок, для баланса — допустимые пределы. Процессную метрику можно наблюдать без порога, если её используют для разбора причин; это укажите прямо. Согласованные условия успешности Пилота должны включать необходимые показатели баланса. На Пилоте ориентиры могут быть уточнены по новым основаниям; сохраняйте прежние и новые значения, причину, решение, версию правил и период применения. При изменении оценки объёма или метода расчёта обеспечьте сопоставимый пересчёт базы и результатов либо отметьте различия периодов. Следуйте этапному порядку, не подменяя проверку результата изменением порога без основания.

5. Связать показатели с данными и оценками

Вместе с ИИ-инженером проверьте путь: первичное событие → задача/MR/запуск → при необходимости оценка, подтверждённая РП → подготовленный набор → повторяемый расчёт. Например, ожидание ревью требует событий запроса и начала содержательной проверки, а не только времени первого сообщения.

Активные затраты людей учитываются отдельно от ожидания и включают помощь, повторные проверки, HOTL и ручное завершение. Оценки идеальных часов и человеческого участия формируются по правилам метрик; способ их подтверждения РП записывается в плане. Сбор опирается на доступные артефакты; план не вводит обязательных таймшитов или ручного отчёта о времени после каждой задачи. Отсутствующие сведения видны как пропуски, а не нули.

6. Проверить сбор на PoC

Настройте ежедневный сбор, ручной запуск за пропущенные дни и пересборку без повторного учёта. Получите пробную недельную сводку, проверьте связи с исходными сведениями и выделенные аномалии оценок. Ограничения малой выборки запишите явно. До Пилота план должен описывать работающий способ наблюдения, а не только список желаемых показателей. Работы и подтверждения — в чек-листе готовности к Пилоту.

7. Вести план и результаты раздельно

План меняют, когда уточняются выборка, показатели, источники или согласованные условия. Дневные и недельные значения сохраняются в журнале задач, недельных сводках и наборах расчёта. Решения об улучшениях принимаются по гайду управления эффективностью. При изменении условий сохраняйте прежнюю версию и период её применения.

План мониторинга личного кабинета Меридиан Сервис

Учебный пример **раннего PoC** по шаблону плана мониторинга. Показывает первый шаг заполнения: какие метрики выбраны, что они означают и какие сведения понадобятся. Исходные значения, численные ориентиры, результаты расчётов и согласования пока не заполнены. Это не план, уже подтверждённый к началу Пилота.

Паспорт проекта и цели внедрения: агентская разработка в личном кабинете. Цель — кратно ускорить поставку функциональности без сопоставимого роста полной стоимости разработки, сохраняя качество. Ниже выбраны показатели сроков, человеческого труда и качества; финансовые показатели пока не заполняются и вывод о полной стоимости по этому набору не делается.

Версия плана, дата применения и ссылка на решение: первичная запись метрик для настройки на PoC. Дата применения проверенной версии и ссылка на согласование появятся после соответствующих работ; их наличие не предполагается заранее.

1. Область и условия наблюдения

Поток работ: задачи развития собственного web/API кабинета в Jira `PORTAL`, изменения в `customer-portal-web` и `customer-portal-api`. Выбираются работы в границах паспорта: срочные задачи без запаса на исправления и задачи с не обеспеченными внешними зависимостями не включаются в сопоставимый поток автоматически.

Процессы и точки измерения: передача задачи агентам, готовность MR к ревью и запрос проверки, начало содержательного ревью, готовность изменения к тестированию и начало проверки тестировщиком, исправления, успешные проверки и приёмка. Точные события сопоставляются с ЖЦ задач и правилами Git; метки начала действий ещё предстоит проверить.

Период наблюдения: на этой стадии — пробы PoC. Плановые сроки этапов заданы в паспорте; для Пилота календарь когорт, отсечки и дополнительное окно наблюдения ещё уточняются. Часовой пояс — Москва.

Условия сопоставимости: на PoC общение с агентами ведёт Никита Логинов, ИИ-инженер «Практики ИИ»; предметные специалисты подключаются через него. Поэтому время его проверки MR и тестов не выдаётся за обычную нагрузку производственной команды. На Пилоте действия выполняет команда продукта; затраты по этим режимам различаются в данных. Версии агентов, моделей, проверки и конфигурация сред связываются с результатами наблюдения.

Ограничения выводов: выбран набор для первичной настройки, доступность всех событий ещё не проверена. Результаты PoC не считаются автоматически исходной базой Пилота. Известные исторические сведения из описания предприятия относятся ко всему потоку развития и исправлений и не подставляются в метрики выбранной области без проверки сопоставимости.

2. Выбранные метрики

На раннем PoC заполнены назначение, определение, предполагаемые границы и источники. **Поля исходной базы, предварительных целевых значений и подтверждённого расчёта пока остаются незаполненными для всех метрик.** Их не заменяют нулями. Перечень уточняется после проб: выбор показателя ещё не доказывает, что он технически вычислим.

M-01 · Медианное время ожидания ревью MR

- **Назначение:** процесс.
- **Связь с целью или решением:** найти задержки между готовностью изменения и началом его проверки, чтобы ускорять поток, а не только работу агента.
- **Определение и единица:** медиана интервалов от доставки запроса ревью готового MR назначенному проверяющему до начала содержательной проверки; рабочие часы по календарю команды. Применяются правила длительности этапов.
- **Проектные границы:** черновой MR не запускает ожидание; каждый повторный запрос ревью образует отдельный интервал. Незавершённые ожидания показываются отдельно с их возрастом, не считаются нулевыми. Время активного ревью не входит в ожидание.
- **Источники и необходимые сведения:** GitLab — готовность MR, запрос проверяющему и действия ревью; Mattermost — доставка обращения. На PoC нужно определить, как достоверно фиксировать начало проверки. Первое сообщение «увидел» не считается началом содержательного ревью; первый видимый комментарий без дополнительной проверки источника тоже не выдаётся за точное время начала.

M-02 · Медианное время ожидания тестирования

- **Назначение:** процесс.
- **Связь с целью или решением:** обнаруживать очередь перед проверкой и отделять её от длительности самих тестов.
- **Определение и единица:** медиана интервалов от готовности нужной ревизии и тестовой среды и доставки запроса тестировщику до начала содержательной проверки; рабочие часы. Основание — длительность этапов.
- **Проектные границы:** если среда ещё не готова, этот интервал относится к подготовке среды, а не ожиданию тестировщика. Очередь автоматического CI, длительность прогона и труд тестировщика не смешиваются с ожиданием человеческой проверки. Повторные запросы и открытые ожидания учитываются явно.
- **Источники и необходимые сведения:** Jira, доставка запроса в чате, GitLab CI и тестовая среда — задача, ревизия, её готовность, запрос и начало тестирования. Одного перехода задачи в «Тестирование» недостаточно, пока не выяснено, означает ли он постановку в очередь или фактическое начало действий.

M-03 · Оценка экономии человеческих трудозатрат по решённым задачам

- **Назначение:** результат.
- **Связь с целью или решением:** проверить, уменьшается ли труд людей на сопоставимый выполненный объём, а не только число ручных действий в интерфейсе агента.
- **Определение и единица:** для решённой агентом задачи — разность идеальных часов выполнения того же состава работ людьми без агентов и активных часов участия людей при её

решении с агентами. Недельный итог — сумма разностей по соответствующей выборке, в человеко-часах; правила расчёта.

- **Проектные границы:** предполагается одинаковая граница — от передачи подготовленной задачи до принятого результата. Включаются уточнения после передачи, помощь инженера, ревью, тестирование, повторные проверки, исправления и интеграция. Время ожидания и работы агентов не вычитается как человеческий труд. Общие работы по развитию навыков учитываются отдельно. Отрицательная экономия не обнуляется. Переданные людям, отменённые и незавершённые задачи с их затратами показываются рядом и не скрываются из оценки потока.
- **Источники и необходимые сведения:** согласованный состав работ, результаты Jira/Git/CI, обращения и действия людей, итоговая идеальная оценка и труд людей. Зарегистрированные фактические часы используются там, где они есть; полученные агентом оценки отмечаются как оценочные, проверяются РП и не называются измеренными фактами. Окончательные границы и проверка оценивания уточняются на PoC.

M-04 · Медианное время от передачи задачи до успешного тестирования

- **Назначение:** результат.
- **Связь с целью или решением:** видеть, ускоряется ли получение проверенного изменения с учётом ожиданий и исправлений.
- **Определение и единица:** календарные часы от подтверждённой передачи готовой задачи агентам до успешного завершения обязательного набора тестов нужной ревизии; TTM до выбранной точки.
- **Проектные границы:** учитываются все необходимые изменения web/API и их проверки. Публикация MR не завершает измерение, тест старой ревизии тоже не подходит. Это время до успешного тестирования, а не до production. Для итогового показателя успешных решений требуется подтверждение исхода задачи по правилам методологии; длительные, открытые и переданные людям задачи раскрываются отдельно.
- **Источники и необходимые сведения:** разрешение взять задачу в Jira или чате, связи задачи с MR, commit/digest, результаты обязательных проверок и последующая приёмка. Состав проверок и способ фиксации передачи задачи предстоит подтвердить на PoC.

M-05 · Доля задач с возвратом с тестирования

- **Назначение:** баланс.
- **Связь с целью или решением:** проверить, не оплачено ли ускорение ростом переделок после тестирования.
- **Определение и единица:** процент задач с хотя бы одним подтверждённым возвратом среди задач с завершённым первым проходом тестирования и полным окном наблюдения; правила B1.
- **Проектные границы:** причина возврата связывается с задачей и проверяемой ревизией. Повторные возвраты не размножают задачу в числителе; проблемы среды отделяются от дефекта изменения. Нельзя сравнивать качество при разном составе проверок без указания этого различия.
- **Источники и необходимые сведения:** история Jira, замечания тестировщика или инженера, отчёт проверки и ссылка на дефект/исправление. На PoC проверяется, какие записи подтверждают именно возврат с тестирования, а не иной переход статуса.

M-06 · Активные часы работы проверяющих

- **Назначение:** баланс.

- **Связь с целью или решением:** убедиться, что ускорение агентов не увеличивает скрытую нагрузку на инженера, разработчиков и тестировщиков.
- **Определение и единица:** сумма активных человеко-часов проверки, отдельно по роли и виду участия; правила В3.
- **Проектные границы:** включаются повторные проверки; ожидание не является активным трудом. На PoC отдельно виден Никита, выполняющий проверки, на Пилоте — фактические участники продукта и помощь инженера. Одно действие не учитывается дважды при совмещении ролей. Это часть затрат М-03, её не прибавляют к ним повторно.
- **Источники и необходимые сведения:** связанные с задачей MR, замечания, прогоны и другие артефакты действий проверяющих. Наблюдаемые и оценочные часы различаются. Новый ручной табель для получения метрики не вводится.

М-07 · Доля задач, переданных от агентов людям

- **Назначение:** процесс.
- **Связь с целью или решением:** видеть, какая часть взятого агентами потока потребовала ручного продолжения, и разбирать связанные ограничения.
- **Определение и единица:** процент задач с подтверждённой передачей человеку среди задач, впервые переданных агентам в работу в той же когорте. Отдельно сохраняется количество передач; это две величины, а не одна смешанная метрика.
- **Проектные границы:** повторные записи одной передачи не создают новые задачи. Труд человека после передачи не исчезает из учёта. Передать задачу можно без обязательного объяснения причины; позднее факт передачи разбирается при управлении потоком.
- **Источники и необходимые сведения:** Jira, сообщения о передаче и журналы запуска агентов; связь с исходной задачей и подтверждённым ручным продолжением. Способ регистрации исхода уточняется на PoC по работе POC-21.

М-08 · Медианное время внутренних ожиданий агентов по причинам

- **Назначение:** процесс; помогает выбирать, устранять ли зависимости, менять параллельность, готовить среды или повышать доступность систем.
- **Определение и единица:** медиана календарных часов завершённых ожиданий **отдельно по каждой причине**, число интервалов, открытые ожидания и их возраст на срезе; расчёт и перекрытия.
- **Обязательные группы:** очередь готовых задач; зависимость/последовательное выполнение; очередь слота Джины; готовность тестовой среды; инфраструктурный отказ. Очереди CI-runner и LLM выделяются при наличии соответствующего препятствия. Неприменимость отдельных причин предстоит подтвердить, а не предположить.
- **События и источники:** таблица ожиданий ЖЦ и события Git/CI: Jira, очередь/журналы Джины, CI, резервирование среды, health check и ошибки инфраструктуры. Точные поля и доступность истории проверяются на PoC. Для первичного набора численные значения ещё не получены.
- **Границы:** очередь готовых задач и ожидания отдельных запусков не складываются автоматически. Отказ общей системы учитывается только у действительно заблокированных действий; для общей задержки задачи пересечения считаются один раз. Простой, активная работа агентов и токены разделяются.

Обязательная детализация HITL

M-01 и M-02 раскрывают очередь ревью и тестирования; дополнительно теми же правилами учитываются запросы уточнения требований и приёмки из ЖЦ задач. Для каждого вида отдельно показываются длительности, открытые запросы и возраст, адресат и покрытие событиями. На позднем PoC обращение к Никите и последующее обращение к специалисту не сливаются в один запрос. M-06 измеряет активный труд и не заменяет эти ожидания. При введении HOTL его отложенное рассмотрение и исправления остаются видимыми, хотя оно не блокирует движение задачи.

3. Первичные сведения и оценочные значения

Первичные источники: целевые Jira `PORTAL`, оба GitLab-репозитория кабинета и CI, проектный канал Mattermost, журналы Джины и тестовых сред. Это перечень предполагаемых источников; чтение нужных полей, полнота истории и связь событий ещё должны быть проверены.

Оценки: идеальные часы предполагается рассчитывать агентом по составу выполненной задачи с проверкой РП; команда может уточнять основания. Часы участия людей собираются из доступных действий и артефактов, с различением зарегистрированных и оценённых величин. Правила и версии оценивания проверяются на PoC и уточняются на Пилоте.

Подтверждение оценок РП: Марина Фомина со стороны продукта и Кира Воронцова со стороны внедрения проверяют оценки в составе общей сводки по своим областям участия. Порядок отбора аномалий и фиксации согласования предстоит уточнить до Пилота. Сами подтверждения ещё не приведены.

Наборы для расчёта: место сохранения и версия навыка/алгоритма ещё не определены. Для каждой величины нужна связь с исходной задачей и периодом; возможность повторить расчёт проверяется пробным сбором.

Полнота: для ожиданий сохраняются незавершённые запросы и их возраст, для задач — исходы и открытая работа, для оценок — наличие и основания. Пропуск не заменяется нулём, незавершённая проверка не объявляется успешной.

Границы затрат: M-03 относится к человеческому труду на задачи, M-06 раскрывает часть этого труда. Токены, работа платформы, общие улучшения и денежные расходы не смешиваются с этими часами. Финансовые значения на данной стадии не заполняются; это ограничивает вывод о полной стоимости внедрения.

4. Сбор и представление результатов

Ежедневный сбор: предусмотрен раз в сутки за прошедший день, по московскому времени; автоматически либо по просьбе агенту. Конкретное время и подтверждённый способ запуска ещё не выбраны.

Пропуски и пересборка: тот же сборщик должен принимать заданные дни или диапазон и исключать повторный учёт. Результат проверки пока не заполнен.

Источники результатов: дневные записи задач, недельные сводки и наборы для повторного расчёта. Конкретные места и формат связей будут определены при настройке; вымышленные URL или готовые отчёты здесь не приводятся.

Недельный разбор: предусмотрен общий разбор РП и ИИ-инженера по гайду процесса. На раннем PoC проверяют доступность событий и правила расчёта; результаты и решения появятся после проб. День и время встречи ещё не зафиксированы.

Итоговая проверка: сроки Пилота берутся из паспорта. К переходу в Пилот нужны исходная база, предварительные ориентиры, работающий сбор и пробная сводка; заполнение только перечня метрик не закрывает этот гейт.

5. Участники и техническое обеспечение

Организуют мониторинг и проверку сводок: Марина Фомина — участие и сведения продукта; Кира Воронцова — работы команды внедрения и общая координация с Мариной. Согласования сохраняются в общей записи.

Настраивает сбор, расчёты и диагностику: Никита Логинов. Денис Пахомов осваивает эти работы при его сопровождении. Подключения инфраструктуры обеспечивает Арсений Дроздов в пределах предоставленных полномочий; версия навыка и результаты проверок ещё не заполнены.

Подтверждают предметные сведения и ограничения: Анна Лебедева — связь с целью и область; Илья Громов — состав требований; Сергей Зотов — ревью и технические ограничения; Вера Мельникова — проверка и возвраты. Их участие на PoC организуется через инженера, самостоятельная работа с агентами не предполагается заранее.

6. Открытые вопросы и изменения

Что уточнить на PoC:

- По POC-21 проверить вычислимость метрик, выбрать окончательный набор, текущие значения и предварительные ориентиры. Организует РП, технические источники проверяет Никита.
- По POC-22 проверить оценивание идеального объёма и участия людей; значения согласуются с РП. Ответственный — Никита.
- По POC-23 проверить суточный сбор и пересборку. Ответственный — Никита.
- По POC-24 проверить пробную недельную сводку и обозначенные ограничения. Организует РП, Никита исправляет выявленные ошибки сбора и расчётов.

Подтверждение готовности к Пилоту: на этой стадии не заполнено. Результаты проверок, согласованный набор и численные ориентиры ещё предстоит получить.

Изменения: первая запись состава метрик. На следующих шагах в этот же документ добавляются подтверждённые источники, исходные и предварительные целевые значения, результаты проб; на Пилоте — уточнения оценивания и ориентиров с сохранением истории.

Паспорт проекта внедрения в продукт

Паспорт фиксирует цель и границы внедрения агентской разработки в одном продукте, конфигурацию, календарь, участников и выделенное время. **Состав команды проекта входит в паспорт**. В одном документе видно, что требуется получить и кто обеспечивает работу со стороны продукта и команды внедрения.

Пример заполнения: личный кабинет Меридиан Сервис.

Шаблон

Скопируйте раздел ниже в рабочий документ проекта. Неизвестные условия обозначайте явно; не придумывайте значения ради заполнения формы.

Паспорт проекта «…»

1. Продукт и цель внедрения

Предприятие и продукт: …

Состояние на дату / версия: …

Программа предприятия, если есть: …

Цель внедрения и исходная проблема: …

Ожидаемый результат и ограничения: …

Согласованные показатели / ссылка на их описание: …

2. Область внедрения

Части продукта и типы задач: …

Какие работы поручаются агентам: …

Что исключено: …

Основания выбора и существенные зависимости: …

Трекер и план выбора задач: …

3. Конфигурация и условия работы

Самостоятельно или с выделенной командой внедрения: …

Платформа, инференс, конвейер и тестовые среды: …

Что меняется между PoC и Пилотом: …

Существенные ограничения и ссылки на подробные описания: …

4. Календарь

Этап	Плановый период	Длительность / адаптация
Подготовка	…	…
PoC	…	…
Пилот	…	…

Контрольные встречи и условия переходов: …

5. Команда проекта

Роль / функция	Участник	Команда / принадлежность	Время по этапам / ограничения
…	…	…	…

Общие участники программы: …

Совмещения, замещение и передача обязанностей: …

Как обеспечивается участие производственной команды: …

6. Ресурсы, ограничения и взаимодействие

Выделенные ресурсы и бюджетные ограничения: …

Как учтена занятость внедрением в планах продукта: …

Рабочие встречи, ответы на обращения и место сохранения решений: …

7. Открытые решения и подтверждения

Что ещё не определено: …

Что согласовано, кем и на каком основании: …

Изменения относительно предыдущей версии: …

Связанные документы и источники: …

Как заполнять

Паспорт начинают на Подготовке. Его ведение организует РП с участием владельца продукта и ИИ-инженера. Документ обновляют при изменении области, назначений и ресурсов; начало Пилота не порождает новый паспорт. Если проект входит в программу предприятия, общие условия и полномочия приводят ссылкой на паспорт программы.

- **Цель.** Объясните, что должно измениться в поставке ПО и почему это нужно продукту. Продуктовая цель вроде увеличения доли самообслуживания не заменяет цель внедрения. На Подготовке цель формулируют содержательно, указывая направление эффекта и известные ограничения, без формул и численных показателей. Метрики, текущие значения и предварительные ориентиры выбирают на PoC в плане мониторинга; на Пилоте уточняют расчёты и при необходимости ориентиры. Позднее в паспорте сохраняют ссылку на согласованные показатели, не копируя формулы и данные расчётов.
- **Область.** Укажите типы задач, части продукта и работы агентов. В паспорте не перечисляют конкретные задачи, в том числе как примеры. На PoC задачи подбирают по ходу работы; при необходимости создают целевые пробы для отработки нужных возможностей. На Пилоте задачи назначают в рабочем порядке либо в конце недели выбирают объём следующей. Подробности остаются в трекере и планах этапов. Отметьте исключения, внешние зависимости и возможность нескольких циклов исправлений до срока поставки. Большой бэклог не означает, что любая задача подходит агентам.
- **Конфигурация.** Отделите самостоятельное внедрение от участия выделенной команды. Сведения о временных и целевых системах приведите кратко со ссылкой на описание инфраструктуры, когда оно появится. Не превращайте паспорт в реестр аккаунтов, сетей или технических проверок.
- **Календарь.** Запишите сроки Подготовки, PoC и Пилота, длительность и существенные адаптации. Контрольные встречи укажите без переноса сюда всех работ. Масштабирование не планируют подробно; его условия рассматривают отдельно.
- **Команда.** Назначайте людей на роли, а не отдельные новые роли для каждой стороны. Обязанности следуют из роли; запишите принадлежность участника, время и замещение. Совмещение РП и ИИ-инженера допустимо. У конкретной работы в плане один ответственный; паспорт не заменяет эти назначения.
- **Взаимодействие и готовность.** На PoC общение с агентами ведёт ИИ-инженер, команда проходит обучение и демонстрацию; при подключении специалистов инженер сопровождает обмен в обе стороны. На Пилоте команда взаимодействует с агентами непосредственно. Подробный маршрут HITL и правила Git/трекера остаются в описаниях процессов.

Согласование паспорта фиксируют фактическим решением и ссылкой на него. Если паспорт ещё содержит несогласованные условия, их нельзя считать выполненными пунктами гейта. Существующая должность человека и его доступ к системе не доказывают выделение времени или необходимых полномочий.

Паспорт проекта «Агентская разработка в личном кабинете Меридиан Сервис»

Учебный пример по шаблону паспорта проекта. Показывает плановые условия начала Подготовки из базы Меридиана; не является подтверждением готовности к PoC.

1. Продукт и цель внедрения

Предприятие и продукт: Меридиан Сервис, личный кабинет заказчика. Web-продукт используется в составе сервисного обслуживания; отдельные лицензии не продаются. Собственная команда владеет репозиториями `customer-portal-web` и `customer-portal-api`.

Состояние на дату / версия: план начала Подготовки с 1 июня 2026 года. Последний описанный исходный срез продукта — 4 мая, показатели за апрель. Production в этом срезе — web/API 2.8.0; обновлённое состояние на 1 июня база не содержит.

Программа предприятия: «Агентская разработка в Меридиан Сервис». Кабинет — первый продукт пилотной волны 2026 года.

Цель внедрения и исходная проблема: кратно ускорить поставку функциональности без сопоставимого роста полной стоимости разработки. Команда имеет очередь фич на несколько месяцев; существующая ёмкость не позволяет реализовать все потребности бизнеса требуемым темпом.

Ожидаемый результат и ограничения: проверить работу агентов в связанном цикле уточнения, реализации, тестирования и исправлений. Сохранить требуемое качество, правила доступа и предметную приёмку. Договорный срок готовности согласования смет для «СеверМаркета» — до 1 августа 2026 года; внедрение само по себе не переносит это обязательство. Рост доли заявок через кабинет с 35% в апреле до 55% в сентябре является целью развития продукта, а не расчётом эффекта агентов.

Согласованные показатели / ссылка на их описание: числовые показатели эффекта пока не заданы в сценарии. В PREP-02 формулируется цель по сути без численных показателей. На PoC предстоит выбрать вычислимые метрики, текущие значения и предварительные ориентиры в плане мониторинга; на Пилоте уточняются расчёты, особенно оценка объёма задач, и при необходимости целевые значения. Согласованный план мониторинга в базе пока отсутствует. Качественная цель не выдаётся за уже утверждённый числовой критерий завершения Пилота.

2. Область внедрения

Части продукта и типы задач: развитие собственных web/API кабинета: интерфейсы самообслуживания, формы и списки, прикладные API, обработка данных и уведомлений, автоматические проверки. Изменения должны иметь проверяемый результат, а необходимые знания и решения быть доступны внутри команды.

Какие работы поручаются агентам: детализация требований, реализация, подготовка и выполнение проверок, исправление замечаний в согласованных границах. На PoC рабочий цикл ведёт ИИ-инженер; область регулярной работы на Пилоте уточняется по результатам проб.

Что исключено: срочные изменения, сроки которых не позволяют пройти несколько циклов проверки и исправлений; работы, основное содержание которых — внешние переговоры и согласования; изменения чужих систем и закрытой конфигурации 1С вне полномочий команды. Изменения с внешними интеграциями включают после обеспечения необходимых зависимостей и способа проверки.

Основания выбора и существенные зависимости: кабинет относится к разработке функциональности, имеет собственную команду, собственный web/API-конвейер и бэклог на несколько месяцев. Работающие интеграции с планированием и учётной системой можно использовать в пределах доступных интерфейсов; новые контракты и поставки соседних команд не считаются обеспеченными заранее.

Трекер и план выбора задач: используется проект Jira `PORTAL`. На Подготовке задаются рамки типов работ. На PoC ИИ-инженер подбирает задачи по результатам проб; при необходимости формулирует специальные задачи для отработки навыков или взаимодействий. На Пилоте задачи назначают из текущего потока либо в конце недели выбирают объём на следующую. Конкретные постановки и назначения остаются в трекере и планах этапов, а не в паспорте.

3. Конфигурация и условия работы

Самостоятельно или с выделенной командой внедрения: с привлечением специализированной компании «Практика ИИ». Компания помогает подготовить инфраструктуру, адаптировать агентов, наладить проверки и измерение результатов, обучить специалистов на задачах продукта. Специалисты кабинета сохраняют предметные решения. Денис Пахомов осваивает роль ИИ-инженера продукта у Никиты Логинова; полная передача сопровождения заранее не предполагается.

Платформа, инференс, конвейер и тестовые среды: выбран общий экземпляр `gina-meridian` в новом отдельном Kubernetes-кластере предприятия; инференс — GLM 5.3 и GLM 5.3 flash от «Практики ИИ». С PoC используются целевые GitLab, Jira, Confluence и Mattermost. У кабинета исходно одна общая `integration`; в проекте создаются feature-среды в продуктовом кластере. План и подключения — в карточке инфраструктуры. На исходном срезе Джина и её подключения ещё не работают.

Что меняется между PoC и Пилотом: целевые системы конвейера сохраняются. Уточняются версии, лимиты, проверяемая область и готовность feature-сред по результатам PoC; изменённые подключения перепроверяются. На PoC общение с агентами ведёт Никита, специалисты подключаются через предварительную проверку материалов и обратной связи. Команда проходит обучение и демонстрацию без требования самостоятельной практики. На Пилоте команда взаимодействует с агентами непосредственно при сопровождении «Практики ИИ».

Существенные ограничения и ссылки на подробные описания: feature-сред и общего механизма сброса данных исходно нет; общая `integration` используется для интеграционных проверок и приёмки. Из 22 E2E шесть нестабильны из-за общих данных или ожиданий. Внедрение включает подготовку изолированных сред, данных, моков и диагностики. Передача материалов «Практике ИИ» допустима только для LLM-вызовов. Техническое устройство продукта описывает исходное состояние, карточка — план, а не подтверждение готовности.

4. Календарь

Этап	Плановый период	Длительность / адаптация
Подготовка	1–21 июня 2026	3 календарные недели
PoC	22 июня – 26 июля 2026	5 недель: онбординг, простые и целевые задачи; пятая неделя на повторные проверки, демонстрацию и готовность к Пилоту
Пилот	27 июля – 18 октября 2026	12 календарных недель

Контрольные встречи и условия переходов: контроль готовности к PoC — 19 июня, к Пилоту — 24 июля; подведение итогов Пилота — 19 октября после периода наблюдения. Это плановые даты: до перехода подтверждаются условия соответствующего чек-листа. Последующее Масштабирование продукта здесь на работы не раскладывается.

5. Команда проекта

В таблице время записано как **Подготовка / PoC / Пилот**, часов в неделю. Резервы выделяются из существующей занятости; это не фактический учёт и не работа сверх недели.

Роль / функция	Участник	Команда / принадлежность	Время по этапам / ограничения
РП со стороны предприятия	Марина Фомина	Руководитель поставки	8 ч/нед. суммарно с координацией программы, не дополнительный резерв
РП со стороны внедрения	Кира Воронцова	Компания «Практика ИИ»	8 / 8 / 8
Владелец продукта	Анна Лебедева	Команда кабинета	4 / 4 / 4
Аналитик	Илья Громов	Команда кабинета	8 / 4 / 8
Архитектура и разработка	Сергей Зотов	Команда кабинета, технический лидер	4 / 2 / 4
Тестировщик	Вера Мельникова	Команда кабинета	4 / 2 / 4; с 1 июня возвращается к полной занятости кабинетом
ИИ-инженер продукта по мере обучения; разработчик	Денис Пахомов	Команда кабинета	8 / 8 / 16; готовая компетенция на старте не предполагается
ИИ-инженер внедрения	Никита Логинов	Компания «Практика ИИ»	40 / 40 / 16
DevOps предприятия	Тимур Агеев	Общая платформа	8 / 4 / 2; инфраструктурное дежурство одну неделю из трёх
DevOps внедрения	Арсений Дроздов	Компания «Практика ИИ»	16 / 8 / 4; не выделяет ресурсы предприятия своим решением

Роль / функция	Участник	Команда / принадлежность	Время по этапам / ограничения
Разработчик backend	Игорь Сазонов	Команда кабинета	Отдельный резерв не задан; время HITL учитывается при уточнении планов
Разработчик frontend	Елена Круглова	Команда кабинета	Отдельный резерв не задан; время HITL не считается нулевым
Тестировщик	Ксения Назарова	Команда кабинета	Отдельный резерв не задан; участие планируется по задачам

Общие участники программы: спонсор Виктор Мельников, руководитель программы Алексей Орлов, методолог Елена Миронова после обучения, инфраструктурные работы организует Евгений Сафонов. Их вклад и время — в паспорте программы, не повторяются как дополнительные часы проекта.

Совмещения, замещение и передача обязанностей: Марина совмещает координацию программы и РП кабинета. Денис остаётся одним из восьми сотрудников продукта, а не новой штатной единицей. При дежурстве Тимура Евгений согласует резервирование работ или замещение Михаилом Даниловым при необходимых правах. Конкретные передаваемые Денису работы уточняются по освоению; сложные случаи остаются у инженера внедрения.

Как обеспечивается участие производственной команды: на PoC предметные специалисты предоставляют сведения и решения, проходят обучение и наблюдают демонстрацию. Их самостоятельная работа с агентами не является условием гейта. При подключении коллег Никита проверяет материалы и обратную связь в обе стороны. На Пилоте участники выполняют действия своей роли непосредственно; у каждой конкретной работы в плане один ответственный.

6. Ресурсы, ограничения и взаимодействие

Выделенные ресурсы и бюджетные ограничения: резервы времени заданы плановым сценарием. Денежный бюджет, стоимость услуг внедрения, договорные условия и подтверждения выделения ресурсов в базе отсутствуют. Участники внедрения не входят в штат предприятия и не увеличивают штатную команду кабинета с восьми до одиннадцати человек.

Как учтена занятость внедрением в планах продукта: обучение Дениса уменьшает обычную плановую разработку примерно с 20 до 18 человеко-дней на полную итерацию в Подготовке/PoC и до 16 на Пилоте при сохранении остальных резервов. Дополнительные затраты уточняются по факту, результаты агентов до подтверждения не добавляются к мощности команды. Расчёт и зависимости — в роадмапе.

Рабочие встречи, ответы на обращения и место сохранения решений: в предприятии есть Mattermost, Jira и Confluence. Канал внедрения, регулярное расписание и конкретные назначения на обработку обращений в исходном сценарии не заданы. Контрольные даты указаны в разделе 4; они не заменяют ежедневного взаимодействия и не доказывают его организацию.

7. Открытые решения и подтверждения

Что ещё не определено: версия поставки, размеры ресурсов и квоты, условия хранения LLM-запросов у поставщика, финансовый бюджет и согласованные показатели эффекта. Канал `portal-ai` предусмотрен в конфигурации, фактическое расписание регулярных встреч ещё не задано. В базе нет обновлённого состояния продукта на 1 июня, результатов установки и подтверждений выделения плановых резервов.

Что согласовано, кем и на каком основании: представлены плановые условия сценария. Протоколов согласования паспорта, приёмки инфраструктуры и перехода к PoC в базе нет. Содержание этого примера не позволяет отметить PC-01 или другие условия готовности выполненными.

Изменения относительно предыдущей версии: предыдущая версия паспорта отсутствует.

Связанные документы и источники: сценарий внедрения, состояние кабинета, бэклог, роадмап, паспорт программы.

Жизненный цикл задач

Карточка сведений описывает общий процесс задач людей и агентов: поступление, выполнение, проверки, возвраты и завершение. Она связывает состояния трекера с правилами Git и задаёт смысл событий для плана мониторинга. Отдельный трекер событий вручную не заполняется: подтверждения остаются в рабочих системах.

Пример: кабинет «Меридиан Сервис».

Шаблон

1. Область и редакция процесса

Продукт, типы задач и границы: … . **Трекер и проект:** … . **Связанные документы:** паспорт …, Git-процесс …, карточка инфраструктуры … .

Состояние описания: действующий процесс / порядок PoC / порядок Пилота; дата начала применения, версия, согласование участников … . Планируемые изменения и подтверждённые настройки отмечаются отдельно.

Что сопоставляем	Действующий процесс	PoC	Пилот
Трекер, проект и выборка задач	…	…	…
Статусы и правила переходов	…	…	…
Кто поручает задачу и разрешает запуск	…	…	…
Вопросы, проверки и приёмка	…	…	…
Какие различия проверить перед Пилотом	—	…	Подтверждение проверки …

2. Статусы и переходы

Статус / состояние	Условие входа и обязательные сведения	Работа и кто её выполняет	Условие выхода и подтверждение
…	…	…	…

Правила готовности задачи: результат, критерии приёмки, зависимости, необходимые материалы и способ проверки … . **Поручение агенту:** разрешённый статус и условия выборки либо поручение в чате, полномочная роль, связь с версией требований … . Повтор поручения не создаёт новый запуск задачи и не меняет дату первого разрешения.

Ограничения параллельности: какие задачи могут идти одновременно, как задаётся порядок зависимых задач и кто разрешает конфликт за общий компонент/среду … . Признак «разрешена агенту» отделяется от «агент начал выполнение».

Возврат, отмена и передача человеку: кто меняет исполнителя, как прекращаются запуски и сохраняются ветки, MR и результаты; как фиксируется исход … . Передать задачу человеку можно по решению команды без обязательного объяснения причин. Отмена и передача не являются успешной поставкой.

3. Участие людей и точки HITL

HITL — запрос решения или действия человека, без которого соответствующая часть работы не продолжается. Состояние трекера само по себе не означает, что человек уже работает. Для правил веток и ревью используется Git-карточка; здесь описываются требования, тестирование и приёмка, а также ссылки на точки ревью.

Контакт / событие	Что предъявляется; что блокируется	PoC: кто отвечает	Пилот: кто отвечает	Как подтверждаются начало работы и результат
Уточнение требований	…	…	…	…
Ревью MR	Ссылка на правило Git …	…	…	…
Тестирование	Ревизия, готовая среда, набор проверок …	…	…	…
Приёмка результата	…	…	…	…

Любой запрос HITL система направляет через чат с наступившим событием и ссылкой на материалы: «MR готов к ревью», «требуется уточнение», «результат готов к проверке». Указываются задача, ревизия и адресат. Ответ и решение доступны агенту. Напоминание не создаёт новую проверку. Запросы, дополнительно возникающие по навыкам агента, сохраняются таким же способом с собственным предметом.

Режим контроля: для каждой проверки HITL / HOTL / автоматическая проверка; согласованная область применения … . HOTL не блокирует движение, но сохраняет работу человека и последующие исправления. Смена режима оформляется решением, а не автоматически при окончании PoC.

Приёмка и UAT

Порядок приёмки выбранной области: принимающий и технические проверяющие …; обязательные проверки …; среда и способ предъявления …; признаки успешного результата … . Эти правила заполняются здесь, отдельная карточка UAT не требуется.

1. Передать комплект для проверки: задача и версия требований, необходимые MR/SHA и артефакты, данные, шаги, результаты автоматических проверок и известные ограничения. Для нескольких репозиториев фиксируется совместимый комплект ревизий.
2. Обеспечить готовую среду и данные. Во время приёмки версия и состояние сохраняются до согласованного изменения. Агенту доступны проверяемая версия и диагностика; это не означает автоматического права менять или сбрасывать общую среду.
3. Направить запрос через чат с событием и ссылками. Отдельно фиксировать готовность среды, доставку запроса, начало содержательной проверки и вердикт. Очередь человека начинается после готовности материалов и среды; подготовка среды учитывается отдельно.

4. Проверить поведение по критериям приёмки. Техническое одобрение кода, успешные тесты, продуктовая приёмка и выпуск сохраняются как разные события с участниками и ревизиями по общему процессу.
5. При дефекте сохранить шаги, ожидаемое и фактическое поведение, ревизию и доказательства; вернуть на исправление и повторную проверку. Новое требование или сбой среды отделить от дефекта изменения. После смены ревизии определить нужные повторные проверки.
6. Сохранить решение принимающего и подтверждения. Успешный прогон старой версии, отсутствие ответа или создание MR не считаются приёмкой. Перевод в следующий статус и выпуск выполняются по описанным условиям процесса.

На PoC материалы и замечания проходят через ИИ-инженера, предметные решения остаются за продуктом. На Пилоте специалисты работают непосредственно, инженер помогает. При временных средах PoC здесь фиксируются различия и проверки целевого порядка Пилота. Если для части контроля согласован HOTL, его находки и исправления сохраняются по правилам этого режима; молчаливого снятия обязательной приёмки нет.

4. Ожидания и их подтверждения

Используйте единые правила измерения ожиданий. Ниже сопоставляются реальные события проекта; это описание источников, а не новый журнал для команды.

Причина	Начало ожидания	Конец ожидания	Подтверждения и условие продолжения
Очередь готовых задач	Разрешено взять готовую задачу	Началось выполнение либо выбрана более конкретная причина ожидания	Поручение / история трекера, журнал планировщика; порядок выбора …
Зависимость или последовательное выполнение	Дальнейшая работа заблокирована названной задачей/изменением	Зависимость устранена	Связи задач, события завершения и разрешения продолжить …
Лимит параллельных запусков	Исполнимый запуск принят в очередь без свободного слота	Слот выделен, запуск стартовал	Очередь платформы, ID запуска, установленный лимит …
Тестовая среда	Нужна среда для конкретной ревизии, она недоступна/занята/ готовится	Нужная ревизия и данные готовы, среда доступна	Резервирование, CI, проверки готовности, ревизия …
Отказ инфраструктуры	Сбой реально блокирует действие задачи	Восстановлено и проверено условие продолжения	Ошибка, инцидент и проверка восстановления; затронутые задачи …
Ожидание HITL	Готовые материалы переданы адресату	Началась содержательная работа человека	Доставка запроса, событие начала проверки / ответа …

Дополнительные причины: очередь CI, ожидание инференса, окна интеграции/выпуска и другие существенные ограничения … . Для каждой — начало, конец, источник и затронутая работа. Ожидание ответа/решения и очередь до начала проверки — разные интервалы.

Доступные сведения: идентификаторы задачи, запроса/попытки, MR и ревизии, причина, время и часовой пояс, адресат либо ресурс, признак блокирования, подтверждения … . Место хранения и способ получения настраивает ИИ-инженер; отсутствующие события отмечаются как пробел, а не выводятся из догадки.

5. Применение и проверка

Кто согласует: РП организует описание; аналитик подтверждает требования, разработчик — выполнение и Git, тестировщик — проверки, ИИ-инженер — применимость агентам и наблюдаемость. Конкретные назначения — в паспорте … .

Что проверено: поручение → выполнение → MR → замечание → исправление → тестирование → приёмка; ожидание ресурса, остановка и передача человеку; ссылки на подтверждения … . Для изменившегося процесса Пилота перечислить дополнительные проверки и результаты … .

Изменения: дата, что изменилось в переходах, точках контроля или источниках, область применения и связь с редакцией плана мониторинга … .

Как заполнять

На Подготовке опишите существующие статусы и фактические действия. Не создавайте новые статусы только для мониторинга: очереди, блокировки и начало работы часто выражаются полями, событиями платформы и CI. Отметьте недоступные подтверждения.

На PoC дополните описание общим порядком людей и агентов, сопоставьте временные и целевые системы. ИИ-инженер выполняет регулярные проверки агентов и при необходимости подключает специалистов через себя. Смена инструментов допустима, но контакты, возвраты и решения должны представлять будущую работу Пилота; самостоятельная работа специалистов на PoC не обязательна.

На PoC заполните план мониторинга: ожидание ревью, тестирования и других HITL, очереди и инфраструктурные задержки входят в обязательный охват. Если причина неприменима, запишите основание; если применима, но события не собираются, запишите ограничение и работу по настройке. Во время Пилота уточняйте правила по фактам, сохраняя версии. Не объединяйте результаты PoC и Пилота без учёта того, кто выполнял проверки и в каких условиях.

Жизненный цикл задач: кабинет «Меридиан Сервис»

Пример заполнения карточки ЖЦ задач для первого внедрения. Действующий процесс взят из описания продукта; правила PoC и Пилота ниже — проектируемый порядок. Даты фактических проверок и результаты согласования появятся после выполнения работ.

1. Область и редакция процесса

Описываются задачи разработки и исправления кабинета в Jira `PORTAL`, связанные с `customer-portal-web` и `customer-portal-api`. Инициативы и инциденты `SUP` не подменяют задачи этого потока. Ветки и ревью — по правилам Git, системы и доступы — по инфраструктуре, измерения — по плану мониторинга.

Что сопоставляем	Действующий процесс	PoC	Пилот
Трекер и выборка	Jira `PORTAL`, выбор Анной и командой	Планируется тот же Jira; выбранные задачи поручает агентам Никита	Тот же Jira; согласованный поток задач, поручение уполномоченного участника
Статусы	Существующая модель ниже	Сохраняется; ожидания — отдельные признаки и события	Та же модель, с уточнениями по результатам проб
Взаимодействие с агентами	Пока не включено в регулярную работу	Никита проверяет понимание постановки, MR и результаты; возвращает замечания	Аналитики, разработчики и тестировщики участвуют непосредственно; Никита помогает
Приёмка и полномочия	Предметные решения остаются у продукта	Никита готовит материалы для предметных решений	Решения в штатном процессе продукта
Что ещё подтвердить	История и доступность нужных событий	Чтение задач и вложений, поручения, раунды проверок, временные признаки ожиданий	Прямую доставку запросов специалистам, права и фактические события целевого маршрута

Использование целевых систем запланировано, но не объявляется проверенным. При временной замене на PoC в эту таблицу добавляются фактические система, проект и соответствие статусов; контакты HITL и возврат замечаний сохраняют смысл Пилота. Данные разных режимов помечаются отдельно.

2. Статусы и переходы

Статус	Условие входа	Работа и участники	Выход и подтверждение
Бэклог	Записана потребность	Анна выбирает приоритет, Илья уточняет содержание	Начато уточнение, запись Jira

Статус	Условие входа	Работа и участники	Выход и подтверждение
Уточнение	Потребность взята в проработку	Илья определяет результат, критерии, права и зависимости	Есть критерии приёмки, нужный контракт и способ проверки
Готово к разработке	Постановка подготовлена	Выбор исполнителя и порядка работ; разрешение агентам фиксируется отдельно	Подтверждён запуск выполнения, не только назначение
В работе	Человек или агент начал выполнение	Изменения, проверки, уточнения и исправления	Полный комплект MR готов к ревью, запрос доставлен
Ревью	Предъявлена готовая ревизия	Ревьюер проверяет код и тесты; очередь отделяется от работы	Замечание → «В работе»; обязательные одобрения и merge → «Тестирование»
Тестирование	Изменение интегрировано для общей проверки	Подготовка среды, очередь тестирования, проверка и приёмка различаются событиями	Подтверждённый дефект → «В работе»; успешные проверки и приёмка Анной → «Готово к выпуску»
Готово к выпуску	Принят проверенный результат	Ожидание подходящего выпуска и его подготовка	Конкретные образы установлены в production, выполнены контрольные действия
Выпущено	Подтверждён production	Текущий исполнитель закрывает задачу	Ссылки на версии и проверку поставки

Блокировка остаётся отдельным признаком с причиной и ссылкой на препятствие. Для отмены сохраняется исход «отменена» с причиной; это не «Выпущено».

Поручение агентам. В примере выбран запуск через проектный канал Mattermost: сообщение содержит Jira-задачу, разрешение взять её и согласованный состав работ. На PoC это делает Никита после проверки понимания задачи агентом; на Пилоте — участник, уполномоченный РП в рамках выбранного Анной потока. Статус «Готово к разработке» сам по себе не разрешает агентам забрать весь бэклог. Время поручения сохраняется, даже если вычисления начались позже.

Параллельность. Независимые задачи могут выполняться одновременно в пределах лимита платформы. Для зависимых изменений web/API и конфликтующих изменений общего контракта в Jira задаётся порядок; Сергей подтверждает техническую зависимость. Зависимость от другой задачи и отсутствие слота запуска — разные причины. Численный лимит и подтверждение работы очереди заполняются после настройки Джины.

Передача человеку. По решению команды Никита или текущий исполнитель прекращает новые запуски агента, сохраняет ветки, MR и результаты, меняет исполнителя и фиксирует время передачи. Причина не обязательна. Затраты и история до и после передачи сохраняются; задача не становится успешным решением агента только из-за последующего ручного завершения.

3. Участие людей и точки HITL

Контакт	Что предъявляется и что блокируется	PoC	Пилот	Подтверждение
Уточнение требований	Конкретный вопрос и затронутая часть задачи	Никита разбирает вопрос; Илья даёт предметное решение при необходимости	Илья отвечает напрямую	Запрос и содержательный ответ в доступной агенту записи
Ревью MR	Готовый код и тесты нужной ревизии; merge заблокирован	Никита проверяет; при позднем подключении коллег проверяет материалы в обе стороны	Сергей/Игорь для API, Елена/Денис для web по распределению ревью	Раунды ревью
Тестирование	Нужные образы, данные и способ проверки; приёмка заблокирована	Никита проводит проверки и формулирует замечания; Вера подключается при необходимости	Вера или Ксения проводит проверку	Событие начала проверки и отчёт по конкретным ревизиям
Приёмка	Проверенный результат и ограничения; выпуск без приёмки не разрешён	Никита готовит материалы, Анна принимает продуктовый результат	Анна принимает в общем процессе	Решение Анны со ссылкой на проверенный результат

Все обращения поступают через проектный чат с событием, задачей и ревизией. На позднем PoC два последовательных рассмотрения — инженер и специалист продукта — учитываются раздельно. Внутри обращения дополнительно различаются очередь до начала работы и само рассмотрение. Демонстрация и обучение специалистов предусмотрены; самостоятельное взаимодействие на PoC не требуется.

Начальный режим обязательных проверок — HITL; автоматический CI его дополняет. HOTL не считается введённым: изменение режима рассматривается на Пилоте по согласованным показателям.

4. Ожидания и их подтверждения

Причина	События в этом примере	Где предполагается получать подтверждения
Очередь готовых задач	Разрешение в Mattermost → начало исполнения либо переход к установленной конкретной причине	Связь сообщения с Jira и журналом планировщика Джины
Зависимость	Записанная блокировка конкретной задачей/контрактом → подтверждённое снятие	Связи Jira, история и решение по техническому порядку
Лимит параллелизма	Исполнимый запуск в очереди Джины → выделение слота и старт	Журнал очереди с ID запуска и причиной задержки

Причина	События в этом примере	Где предполагается получать подтверждения
Тестовая среда	Требуется проверка ревизии → получена нужная среда с данными и успешной проверкой готовности	Резервирование integration или feature-среды, CI, digest/SHA
Инфраструктурный отказ	Действие остановлено ошибкой Git/Jira/LLM/среды → повторная проверка подтверждает восстановление	Логи Джины, CI и инфраструктуры; связь с затронутыми задачами
HITL	Материалы готовы и запрос доставлен → человек приступил к содержательной работе	MR, Mattermost, начало проверки/ответа; достоверный способ фиксации уточняется на PoC

На исходной общей integration-среде возможна очередь из-за других проверок и общих данных. Feature-среды предусмотрены проектом инфраструктуры, но до их проверки не считаются доступными. Нельзя ждать человека «на тестирование», пока фактически готовится среда: эти отрезки разделяются. При заранее отправленном уведомлении очередь человека начинается только после готовности материалов и среды.

Очередь CI и ожидание LLM уточняются по доступным событиям отдельных запусков. Внутренние ожидания считаются в календарном времени; HITL дополнительно — по рабочему календарю адресата. Рабочий календарь, включая исключения, предстоит подтвердить на PoC.

Одно ожидание связывается с задачей, попыткой, причиной и ресурсом/адресатом; история берётся из систем автоматически. Если старт ревью нельзя установить, он остаётся неизвестным: время до первого комментария не выдаётся за очередь. Открытые ожидания показываются с возрастом. Перекрытия обрабатываются по правилам расчёта.

5. Применение и проверка

Марина организует согласование со стороны продукта, Кира — работы внедрения. Илья подтверждает постановку и уточнения, Сергей — технический маршрут, Вера — проверки; Никита настраивает агентские действия и сбор. Денис осваивает сопровождение при его участии.

На PoC предстоит проверить поручение, полный раунд исправления MR, тестирование и приёмку, очередь ресурса и передачу человеку. Перед Пилотом дополнительно проверяются изменившиеся подключения и прямые контакты специалистов с агентами. Ссылки на эти проверки, версия процесса и даты применения пока не заполнены. Изменения источников отражаются одновременно в плане мониторинга.

Чек-лист готовности к PoC

GATE-01 · Продуктовая команда · Переход «Подготовка → PoC».

Гейт проверяет документы, проведённые мероприятия и технические результаты Подготовки для согласованного перечня задач PoC. Обучение агентов, проверка их возможностей и доведение мониторинга до готовности выполняются на PoC. В PREP-02 формулируют цель по сути, без формул и численных показателей; метрики и ориентиры выбирают на PoC.

Работы Подготовки.

Условия перехода

- [ ] **PC-01. Паспорт проекта заполнен и согласован.** В паспорте указаны цель, ожидаемое изменение, границы внедрения, участники решений, ограничения качества, рисков и затрат. Сохранены согласование спонсора и дата. Работа: PREP-02.
- [ ] **PC-02. Команда назначена, вводный семинар проведён.** Заполнен раздел «Команда проекта» паспорта с ролями и выделенным временем; записаны договорённостях проекта. В календаре есть ежедневные встречи и еженедельная встреча РП со спонсором; закреплён ответ в течение двух часов рабочего времени. Сохранены презентация, список участников, протокол семинара и отметки выполнения учебного упражнения по обращению агента и ответу человека. Работы: PREP-01, PREP-05, PREP-06.
- [ ] **PC-03. Карточка инфраструктуры заполнена и согласована.** В единой карточке описаны конфигурация инференса, ИИ-платформы, инструментов конвейера разработки и тестовых сред; размещение систем и сетевая связность. Перечислены необходимые учётные записи и права **людям, системам и ИИ-платформе**. Сохранены согласования технических специалистов и ИБ. Работа: PREP-07.
- [ ] **PC-04. Инфраструктура и доступы подготовлены и проверены.** По карточке инфраструктуры системы развёрнуты или подключены, ресурсы выделены, сетевая связность настроена. Учётные записи и права предоставлены **людям, системам и ИИ-платформе**. В этой же карточке записаны успешные проверки сети, аутентификации и операций из соответствующих рабочих контуров, технический прогон существующей ревизии и итог готовности. Обязательных незавершённых работ нет. Работы: PREP-08, PREP-09, PREP-10, PREP-11, PREP-12, PREP-13, PREP-14, PREP-27, PREP-30.
- [ ] **PC-05. Задачи для PoC выбраны и доступны агентам.** После проверки инфраструктуры в целевом трекере выбраны задачи PoC и назначены участники их ведения. Сохранены ссылки на задачи или фильтр трекера по PREP-03. Из рабочего контура агентов проверено чтение описаний и необходимых вложений. Отдельные карточки методологии не заполняются. Работы: PREP-03, PREP-17.
- [ ] **PC-06. Описаны действующий жизненный цикл задач и Git.** Заполнены статусная модель задач и Git-процесс: состояния и переходы, ветки, MR, проверки, ревью и merge. Описания подтверждены участниками соответствующих действий. Подробные правила взаимодействия с агентами оформляются на PoC перед Пилотом. Работы: PREP-15, PREP-16.

- [ ] **РС-07. Сроки и работы PoC согласованы.** В паспорте проекта указаны сроки и контрольные встречи. Для работ по чек-листу готовности к Пилоту в рабочем трекере закреплены исполнители и результаты; порядок онбординга допускает пробы и доработки с сохранением навыков, памяти и подтверждений. Согласование РП, владельца продукта и ИИ-инженера сохранено. Работа: PREP-04.

Решение о переходе

РП готовит материалы и проводит встречу — PREP-28. Спонсор принимает решение — PREP-29.

- Решение спонсора: **начать PoC / продолжить Подготовку**.
- Версия перечня задач и разрешённая область PoC: …
- Дата начала PoC либо дата повторной проверки: …
- Ссылка на протокол и дата решения: …
- Незавершённые работы, если Подготовка продолжается: …

При незакрытом обязательном пункте PoC не начинается. Если команда сокращает перечень задач, новая версия согласуется и заново проверяется по этому чек-листу. План выполнить обязательную работу позднее не является подтверждением готовности.

Работы на этапе Подготовки

Работы выведены из результатов гейта и распределены по пяти процессам. У каждой работы один ответственный; в проекте на эту роль назначается конкретный человек. Соисполнители указаны вместе с их вкладом. Если человек совмещает роли, это фиксируется в карточке команды.

РП ведёт работы в трекере на основе этого перечня, сохраняя ID, ответственного, срок, статус и ссылку на результат. Сроки этапов и состав команды задаются паспортом проекта. Пустая карточка или заявка на доступ не закрывает работу. Уже существующий результат проверяется и подключается ссылкой.

Подготовку начинают со сбора команды, оформления паспорта и описания действующих процессов задач и Git. В PREP-02 формулируют содержательную цель без формул, выбранных метрик и численных ориентиров. Границы продукта и типы работ из паспорта служат основанием для подготовки инфраструктуры: заполняют её карточку, настраивают системы, связность и доступы. ИИ-инженер подтверждает техническую готовность, РП организует личную проверку доступов. После этого в целевом трекере выбирают задачи PoC; ссылки и подтверждения чтения сохраняют в материалах проверки. Дополнительные инфраструктурные условия обеспечивают до включения зависимой задачи. Сроки и контрольные встречи записывают в паспорте, работы по чек-листу готовности к Пилоту назначают в трекере. Конкретные пробы онбординга ИИ-инженер подбирает на PoC, разбирая ошибки, дорабатывая навыки и память и повторяя проверки. Участники проходят вводный семинар. Затем РП готовит материалы и проводит встречу по гейту, на которой спонсор принимает решение о переходе. На PoC уточняют метрики и исходную базу, заполняют план мониторинга и проверяют сбор к Пилоту.

В предшественниках указаны непосредственные зависимости. Работы, уже связанные через других предшественников, повторно не перечисляются; несколько ссылок обозначают независимые ветви подготовки результата. Условная работа учитывается только при её применимости. В конце описания каждой работы указаны предшественники — зависимости завершения. Собирать сведения и

выполнять независимые части можно параллельно. Для встречи по гейту зависимости означают получение статуса каждой работы, в том числе незавершённой: отрицательное решение не требует предварительно закрыть все задачи.

Управление внедрением

Работа и результат	Ответственный и соисполнители
PREP-02. Оформить цели и границы внедрения В паспорте проекта содержательно описаны необходимое изменение в производстве ПО, его смысл для продукта, границы внедрения и ограничения качества, рисков и затрат. Формулы, метрики и численные целевые значения на Подготовке не задаются. Предложение обсуждено с владельцем продукта и согласовано спонсором. Метрики, текущие значения и предварительные ориентиры выбираются на PoC по этапному порядку. **Процесс:** Управление внедрением. **Предшественники:** PREP-01.	**РП** Владелец продукта — формулирует ожидаемое изменение; ИИ-инженер — указывает технические ограничения и вопросы для проверки на PoC; спонсор — согласует предложение.
PREP-03. Выбрать и назначить задачи PoC в целевом трекере После проверки инфраструктуры в целевом трекере выбраны задачи PoC и назначены участники их ведения. Ссылки на выбранные задачи или фильтр PoC сохранены в результатах работы; состав согласован владельцем продукта. Отдельные карточки и копии описаний не создаются. . **Процесс:** Управление внедрением. **Предшественники:** PREP-30.	**Владелец продукта** Аналитик — отбирает задачи из бэклога и сохраняет ссылки; ИИ-инженер — проверяет соответствие подготовленным условиям PoC.
PREP-04. Согласовать сроки и работы PoC В паспорте актуализированы сроки и контрольные встречи. Работы по чек-листу готовности к Пилоту назначены в рабочем трекере: один ответственный, результат, зависимости и срок. В соответствующих работах указан порядок проб и доработок при онбординге, сохранения навыков, памяти и подтверждений. Сроки и назначения согласованы РП, владельцем продукта и ИИ-инженером. **Процесс:** Управление внедрением. **Предшественники:** PREP-05, PREP-17.	**РП** ИИ-инженер — определяет порядок онбординга и проверки возможностей; владелец продукта — подтверждает задачи и участие специалистов; тестировщик — уточняет направления проверок.

Работа и результат	Ответственный и соисполнители
PREP-28. Подготовить материалы и провести встречу по гейту Подготовлен заполненный чек-лист со ссылками на результаты работ. Участникам заранее направлены материалы и повестка; встреча проведена, результаты проверки и выступления записаны в протокол. Незакрытые пункты показаны явно; решение спонсора фиксируется в PREP-29. **Процесс:** Управление внедрением. **Предшественники:** Состояние работ по PC-01–07.	**РП** ИИ-инженер — представляет технические проверки; аналитик — перечень задач и комплект материалов; DevOps — протокол готовности инфраструктуры.
PREP-29. Принять решение о переходе к PoC В протоколе записано решение спонсора: «начать PoC» с областью и датой старта либо «продолжить Подготовку» с перечнем незавершённых работ и датой повторного рассмотрения. При незакрытом обязательном пункте старт не разрешается. **Процесс:** Управление внедрением. **Предшественники:** PREP-28.	**Спонсор** РП — представляет сводку готовности и фиксирует решение; владелец продукта — подтверждает область задач и участие команды; ИИ-инженер — отвечает по техническим ограничениям.

Подготовка и вовлечение участников

Работа и результат	Ответственный и соисполнители
PREP-01. Собрать команду внедрения Заполнен раздел «Команда проекта» паспорта: спонсор, РП, владелец продукта, ИИ-инженер, аналитик и специалисты для задач PoC; у каждого указаны роль и выделенное время. Назначения подтверждены руководителями участников. **Процесс:** Подготовка и вовлечение участников. **Предшественники:** Решение готовить PoC.	**РП** Владелец продукта — предлагает специалистов продукта и подтверждает выделение их времени.
PREP-05. Организовать участие и коммуникации В договорённостях проекта по назначениям PREP-01 для действий с требованиями, MR, тестами и приёмкой записаны доступность и замещения. События обращений и необходимые роли описываются в ЖЦ задач и Git; здесь закреплены конкретные исполнители и время. Созданы рабочий чат, ежедневные встречи и еженедельная встреча РП со спонсором; записано правило ответа в течение двух часов рабочего времени. **Процесс:** Подготовка и вовлечение участников. **Предшественники:** PREP-02.	**РП** Владелец продукта — подтверждает доступность специалистов; ИИ-инженер — перечисляет обращения, требующие участия людей.

Работа и результат	Ответственный и соисполнители
PREP-06. Провести вводный семинар для участников PoC Сохранены презентация, список участников и протокол семинара. На учебном примере разобраны постановка, вопрос агента, ответ человека и разбор результата вместе с ИИ-инженером по действующим процессам продукта. Участники выполнили учебное упражнение: получили обращение в рабочем чате и дали ответ в своей роли; выполнение отмечено в протоколе. Для отсутствовавших проведён повторный разбор. **Процесс:** Подготовка и вовлечение участников. **Предшественники:** PREP-05, PREP-16.	**ИИ-инженер** РП — организует участие и сохраняет протокол; аналитик, разработчик и тестировщик — выполняют упражнение со своими действиями.

Обеспечение технической готовности

Работа и результат	Ответственный и соисполнители
PREP-07. Оформить и согласовать карточку инфраструктуры В карточке инфраструктуры определены конфигурация инференса, ИИ-платформы, инструментов конвейера разработки и тестовых сред; размещение каждой системы и сетевая связность. Перечислены необходимые учётные записи и права людям, системам и ИИ-платформе. Согласования сохранены; фактическое предоставление и проверка выполняются следующими работами. **Подсказка для назначения:** назначить ИИ-инженера, который сводит требования платформы и продукта; он может быть участником команды внедрения или продукта. DevOps с полномочиями в инфраструктуре продукта описывает ресурсы, сеть и доступы; DevOps команды внедрения при его участии уточняет требования установки платформы. Один человек может совмещать эти вклады. Конкретные участники и их части фиксируются в карточке команды и проектном плане. **Процесс:** Обеспечение технической готовности. **Предшественники:** PREP-16.	**ИИ-инженер** DevOps — описывает ресурсы, размещение и связи.; Архитектор — при необходимости уточняет архитектурные ограничения размещения и связей, если они затрагивают выбранную область.

Работа и результат	Ответственный и соисполнители
PREP-08. Развернуть или подключить платформу Для on-prem выполнена установка по документации поставки; для облака или действующего экземпляра создан проект. В карточке инфраструктуры записаны адрес, версия и результаты проверки работоспособности. конфигурация платформы и проверка подключений. **Подсказка для назначения:** развёртывание выполняет DevOps, знающий поставку платформы; это может быть участник команды внедрения или продукта. В on-prem ресурсы и доступ для установки обычно обеспечивает уполномоченный DevOps со стороны продукта/ предприятия. Если это разные люди, в адаптированном плане выделяются предоставление ресурсов и установка с отдельными ответственными и результатами; установка зависит от предоставления ресурсов и доступа. Для облака или действующего экземпляра назначается участник с полномочиями создать или подключить проект. **Процесс:** Обеспечение технической готовности. **Предшественники:** PREP-07.	**DevOps** ИИ-инженер — проверяет доступность функций для PoC.
PREP-09. Подключить и проверить инференс В карточке инфраструктуры указаны подключения, лимиты и условия использования. Из контура платформы выполнены проверочные вызовы основной текстовой модели и модели зрения; одна модель может выполнять обе функции. Для остальных назначений записана применимость к типам работ в выбранной области. **Процесс:** Обеспечение технической готовности. **Предшественники:** PREP-08.	**ИИ-инженер** DevOps — обеспечивает сетевой доступ.
PREP-10. Организовать хранение материалов внедрения Создан репозиторий внедрения с методологией и указателем рабочих документов; определено место контекста и инструкций агентов. Проверены чтение и запись инженером. **Процесс:** Обеспечение технической готовности. **Предшественники:** PREP-07.	**ИИ-инженер**

Работа и результат	Ответственный и соисполнители
PREP-11. Настроить связи и доступы системам и ИИ-платформе По карточке инфраструктуры настроены сети и созданы служебные учётные записи. Системам конвейера и компонентам ИИ-платформы предоставлены необходимые права на инференс, Git, трекер, CI/CD, анализ кода, артефакты, среды, чат и документацию. Для каждого используемого соединения проверены сеть, аутентификация и операции из фактического контура; результаты внесены в ту же карточку. Чтение выбранных задач дополнительно проверяется в PREP-17. **Процесс:** Обеспечение технической готовности. **Предшественники:** PREP-08, PREP-10.	**DevOps** ИИ-инженер — проверяет подключения компонентов платформы.
PREP-12. Подготовить рабочие места и доступы людям По карточке инфраструктуры участникам выданы аккаунты и роли в необходимых системах. Настроены их рабочие подключения, ИИ-инженеру предоставлен локальный агент или рабочее место на платформе. В карточке записаны выданные доступы и настроенные рабочие подключения; выполнение участниками проверочных действий подтверждается в PREP-30. **Процесс:** Обеспечение технической готовности. **Предшественники:** PREP-08.	**DevOps** РП — предоставляет список участников и необходимые им права; ИИ-инженер — уточняет требования к своему рабочему окружению.
PREP-13. Проверить технический путь сборки и развёртывания В карточке инфраструктуры описан и выполнен технический прогон на существующей ревизии: сборка, публикация артефакта, деплой по неизменяемому ID, контрольный запуск проверок и получение логов. Сохранены команды, ожидаемые результаты и протокол. Это проверка инфраструктуры инженером, а не заранее выбранная задача онбординга. **Процесс:** Обеспечение технической готовности. **Предшественники:** PREP-11.	**DevOps** Разработчик — предоставляет ревизию, команды и ожидаемые результаты проверок; ИИ-инженер — проверяет выполнение из рабочего контура платформы.
PREP-14. Подготовить тестовые среды продукта По области внедрения в карточке инфраструктуры записаны состав сред и результаты проверок: развёртывание, тестовые данные, сброс БД, диагностика, E2E, требуемые моки, эмуляторы оборудования и виртуальное время. Проверки выполняются на существующей версии продукта. Требования к средам. **Процесс:** Обеспечение технической готовности. **Предшественники:** PREP-13.	**DevOps** Разработчик — добавляет механизмы управления зависимостями, данными и временем; тестировщик — готовит данные и проверяет воспроизводимость сценариев; ИИ-инженер — проверяет доступность управления из контура платформы.

Работа и результат	Ответственный и соисполнители
PREP-27. Подтвердить готовность систем и связности В карточке инфраструктуры для обязательных компонентов и соединений записаны успешные проверки. Подтверждены работа инференса, ИИ-платформы, конвейера и тестовых сред, сетевая связность и доступы систем и ИИ-платформы. Сохранены ссылки на технический прогон существующей ревизии; обязательных незавершённых технических работ нет. **Процесс:** Обеспечение технической готовности. **Предшественники:** PREP-09, PREP-14.	**ИИ-инженер** DevOps — устраняет выявленные проблемы систем, связности и служебных доступов; тестировщик — подтверждает результаты проверок тестовых сред.
PREP-30. Подтвердить доступы участников команды РП организовал проверку каждым участником доступов, предусмотренных карточкой инфраструктуры. Каждый вошёл со своего рабочего места под своей учётной записью и выполнил необходимые для своей роли действия в используемых системах. Результат, дата и подтверждение участника записаны в карточке; все обязательные проверки успешны. **Процесс:** Обеспечение технической готовности. **Предшественники:** PREP-12, PREP-27.	**РП** Участники команды — лично выполняют проверочные действия и подтверждают результат; DevOps — устраняет выявленные проблемы доступов.
PREP-17. Проверить чтение выбранных задач агентами Для каждой выбранной задачи проверено чтение описания и необходимых вложений из контура агента через фактически используемое подключение к трекеру. В результатах проверки по PC-05 сохранены ссылки на задачи и подтверждения чтения; при ошибке доступ исправлен и проверка повторена. Выбор задач. **Процесс:** Обеспечение технической готовности. **Предшественники:** PREP-03.	**ИИ-инженер** Аналитик — сверяет прочитанное содержание с задачей и вложениями.

Встраивание агентов в процессы разработки

Работа и результат	Ответственный и соисполнители
PREP-15. Описать общий процесс задач Заполнена и согласована статусная модель: статусы, переходы, условия начала и завершения, поля и участники действий для выбранных типов задач PoC. Как описать процесс. **Процесс:** Встраивание агентов в процессы разработки. **Предшественники:** PREP-02.	**РП** Аналитик — описывает постановку и уточнения; разработчик — выполнение и передачу изменений; тестировщик — проверки и возвраты.

Работа и результат	Ответственный и соисполнители
PREP-16. Описать общий Git-процесс Заполнен и согласован Git-процесс: базовые и рабочие ветки, целевые ветки MR, проверки, ревью, исправления и merge. Переходы согласованы со статусной моделью задач. **Процесс:** Встраивание агентов в процессы разработки. **Предшественники:** PREP-15.	**Разработчик** DevOps — описывает CI и ограничения прав; ИИ-инженер — проверяет применимость операций к агентам.

На Подготовке описываются действующие процессы задач и Git. ИИ-инженер использует их при пробах на PoC и организует необходимые обращения и проверки. Подробный маршрут HITL, порядок приёмки и регламент остановки/передачи оформляются по результатам PoC перед Пилотом.

Управление эффективностью работы команды и агентов

Порядок взаимодействия и регулярных разборов — в гайде процесса.

Практические работы этого процесса выполняются на PoC: пробы, уточнение контекста, доработка навыков и памяти, разбор ошибок и проверка улучшений. На Подготовке сроки задаются в паспорте, работы и результаты берутся из чек-листа готовности к Пилоту и назначаются в PREP-04.

Чек-лист готовности к Пилоту

GATE-02 · Продуктовая команда · Переход «PoC → Пилот».

Гейт проверяет, можно ли передать выбранный поток текущих задач продуктовой команде для регулярной работы с агентами. Возможности агентов подтверждаются результатами PoC, а готовность инфраструктуры и рабочего процесса — проверками в контуре Пилота. Успешный PoC не отменяет проверки новых подключений и не означает автоматического снятия HITL.

Работы PoC · Предыдущий гейт — готовность к PoC.

Условия перехода

- [ ] **PL-01. Подтверждены результаты PoC и границы Пилота.** В журнале задач, журнале улучшений и журнале показателей сохранены ссылки на решённые задачи, изменения, проверки и заключения специалистов. В паспорте проекта перечислены включённые и исключённые типы задач и работы агентов; точки HITL и проверки описаны в ЖЦ задач и правилах Git. Переход к HOTL рассматривается уже на Пилоте по результатам наблюдения за метриками. Для включённой области есть подтверждения из PoC; ограничения и незакрытые замечания перечислены, обязательных для неё исправлений не осталось. Работы: POC-19, POC-01, POC-18, POC-20.

- [ ] **PL-02. Карточка инфраструктуры уточнена для Пилота.** В карточке инфраструктуры указаны системы и среды Пилота, их размещение, связи, ресурсы, учётные записи и права людям, системам и ИИ-платформе. Отмечено, что сохраняется от PoC, что заменяется или подключается заново. Записаны согласования технических решений и ИБ; если изменений нет, это подтверждено в карточке инфраструктуры. Работы: POC-07.

- [ ] **PL-03. Инфраструктура и доступы Пилота проверены.** В карточке инфраструктуры сохранены результаты проверок подключений и операций из рабочих контуров Пилота, проверок доступов участников, сборки, развёртывания и тестов. Тестовая среда поддерживает e2e-проверки выбранной области и воспроизводит необходимые условия продуктивной среды. Все необходимые подключения выполнены; заявок или планов подключения вместо результатов нет. Работы: POC-08, POC-09, POC-10, POC-11, POC-12.

- [ ] **PL-04. Согласован и проверен общий процесс работы людей и агентов.** Уточнены жизненный цикл задач и Git-процесс. В них и связанных правилах описаны запуск агентом из статуса трекера либо по поручению в чате, точки HITL, проверяющие, приёмка и способ передачи задачи человеку по решению команды. Любой запрос HITL поступает через чат с указанием события и ссылкой на материалы. В описаниях сопоставлены точки HITL и другие контакты людей и агентов на PoC и Пилоте; для каждого есть ссылка на проверку. Сохранён сквозной прогон в контуре Пилота, включая исправление замечания, пробу остановки/передачи человеку и взаимодействия, не проверенные на PoC. Работы: POC-13, POC-14, POC-15, POC-16, POC-17.

- [ ] **PL-05. Команда подготовлена к работе на Пилоте.** Обновлён раздел «Команда проекта» паспорта: назначены ведущие задач, проверяющие, замещающие и ИИ-инженер, подтверждено выделенное время. Сохранены материалы обучения, состав участников и ссылка на демонстрацию ИИ-инженера на задачах продукта либо её протокол. В материалах описаны новый процесс и действия участников; продемонстрированы постановка и уточнение задачи,

ревью, возврат замечаний и повторная проверка исправлений. Самостоятельная работа с агентами и обязательная практическая проба на PoC не требуются. В календаре стоят ежедневные встречи и еженедельная встреча РП со спонсором; закреплён ответ в течение двух часов рабочего времени. Работы: POC-02, POC-06.

- [ ] **PL-06. Метрики и исходная база подготовлены.** В плане мониторинга записаны метрики результата, процесса и баланса, связь с целями, источники, правила расчёта и согласованные ориентиры, включая учёт задач, переданных от агентов людям, ожиданий HITL и внутренних ожиданий агентов. Сохранены оценочные значения для сравнения, их основания и подтверждение РП; указаны период, область и ограничения сравнения. Показатели не выдают отсутствие данных за нулевые затраты; человеческая работа по проверкам, помощи и исправлениям включена в учёт. Работы: POC-21, POC-22.

- [ ] **PL-07. Сбор сведений и недельная отчётность проверены.** Сохранены результаты пробного ежедневного сбора и недельная сводка, проверенная командой. Сбор охватывает используемые трекер, Git, чат, журналы агентов и тестовых сред. Проверены повторный сбор без двойного учёта и сбор пропущенных дней. В плане мониторинга указан запуск раз в сутки: автоматически либо по запросу агенту за прошедший день или заданные дни. Работы: POC-23, POC-24.

- [ ] **PL-08. Условия и работы Пилота согласованы.** В паспорте проекта указаны область и поток задач, сроки, ресурсы, участники и контрольные встречи. План мониторинга задаёт согласованные показатели; применяются условия завершения Пилота. Ближайшие работы и ссылка на бэклог или фильтр первой недели сохранены в трекере. Есть согласования РП, владельца продукта и ИИ-инженера; условия учитывают проверенный контур и ограничения PoC. Работы: POC-03.

Решение о переходе

РП готовит материалы и проводит встречу — POC-04. Спонсор принимает решение — POC-05.

- Решение спонсора: **начать Пилот / продолжить PoC**.
- Разрешённая область, режимы проверки и версии паспорта, процессов и плана мониторинга: …
- Дата начала Пилота либо дата повторной проверки: …
- Ссылка на протокол и дата решения: …
- Незавершённые работы, если PoC продолжается: …

При незакрытом обязательном пункте Пилот не начинается. Если область сокращают, её новую версию согласуют и повторно проверяют по этому чек-листу. Улучшения за пределами разрешённой области могут продолжаться отдельно; план будущих работ не заменяет готовность выбранного потока.

Работы на этапе PoC

Работы распределены по тем же пяти процессам, что и Подготовка. У каждой работы один ответственный; в проекте назначается конкретный человек. Соисполнители указаны вместе с их вкладом. РП ведёт сроки, статусы и ссылки на результаты в плане или трекере; отдельные карточки задач методологии не требуются.

На PoC ИИ-инженер через пробы и доработки адаптирует агентов к продукту, а специалисты проверяют результаты. По подтверждённым возможностям выбирают область Пилота. Она определяет необходимые изменения инфраструктуры, процессы, участие людей и мониторинг. Если на PoC использовали временные Git, трекер или среды и упрощённый процесс, для Пилота уточняют карточку инфраструктуры, подключают целевые системы и повторяют зависимые проверки. При сохранении контура проверяют его достаточность для выбранного потока; повторная установка систем не требуется. Команда проходит обучение новому процессу и наблюдает демонстрацию работы ИИ-инженера на задачах PoC. Временные системы и упрощения допустимы, но процесс PoC должен воспроизводить предусмотренные для Пилота точки HITL и другие контакты людей и агентов: поручение работы, вопросы и ответы, ревью, возврат замечаний, приёмку и передачу человеку. При уточнении процесса Пилота сопоставляют эти взаимодействия с проверенными на PoC; недостающие проверяют дополнительно. Готовность новых подключений и рабочего маршрута проверяют в контуре Пилота отдельно; повторять обучение только из-за смены инструментов не требуется; различия процесса доводят до участников. Затем проверяют сбор сведений, согласуют план и выносят решение на встречу со спонсором.

В предшественниках указаны непосредственные зависимости. Работы, уже связанные через других предшественников, повторно не перечисляются; несколько ссылок обозначают независимые ветви подготовки результата. Условная работа учитывается только при её применимости. Предшественники в каждой работе — зависимости её завершения; независимые части можно выполнять параллельно. Для встречи по гейту нужен статус всех работ, включая незавершённые: решение продолжить PoC можно принять без их закрытия.

Управление внедрением

Работа и результат	Ответственный и соисполнители
POC-01. Согласовать область и режимы работы Пилота В паспорте проекта перечислены поток и типы задач, работы агентов, исключения и ограничения на старте Пилота. Точки HITL и проверки указаны в ЖЦ задач и правилах Git. Переход к HOTL рассматривается на Пилоте по метрикам; его подготовка и проба не входят в этот гейт. Для каждого включённого вида работ приведены подтверждения PoC. Новый тип задач либо новый режим без подтверждений в стартовую область не включён. **Процесс:** Управление внедрением. **Предшественники:** POC-19	**Владелец продукта** ИИ-инженер — указывает подтверждённые возможности; разработчик и тестировщик — подтверждают достаточность проверок; РП — согласует границы внедрения.
POC-03. Согласовать условия и работы Пилота Актуализирован паспорт проекта: сроки, поток, ресурсы, участники и контрольные встречи. Показатели заданы в плане мониторинга; применяются критерии завершения Пилота. Ближайшие работы с ответственными и выборка продуктовых задач первой недели сохранены в трекере. Все обязательные условия старта обеспечены. Работы самого Масштабирования не планируются. Сохранены согласования. **Процесс:** Управление внедрением. **Предшественники:** POC-06, POC-17, POC-24.	**РП** Владелец продукта — подтверждает поток и приоритет; ИИ-инженер — сопровождение и технические условия.

Работа и результат	Ответственный и соисполнители
POC-04. Подготовить материалы и провести встречу по гейту Собраны статусы PL-01–08 и ссылки на результаты работ, оставшиеся ограничения и предложение о начале Пилота либо продолжении PoC. Встреча со спонсором проведена, материалы представлены. **Процесс:** Управление внедрением. **Предшественники:** POC-03.	**РП** ИИ-инженер — представляет результаты и технические ограничения; владелец продукта — подтверждает область и участие команды.
POC-05. Принять решение о переходе к Пилоту На встрече рассмотрены результаты PoC и готовность выбранного потока. В протоколе записаны решение, разрешённая область и режимы, дата старта либо незавершённые работы и дата повторной проверки. **Процесс:** Управление внедрением. **Предшественники:** POC-04	**Спонсор** РП — сохраняет протокол и отражает решение в плане.

Подготовка и вовлечение участников

Работа и результат	Ответственный и соисполнители
POC-02. Подтвердить состав команды и время на Пилот В разделе «Команда проекта» паспорта обновлены роли, участники, замещения и время на ведение задач, HITL и сопровождение. В календарь внесены ежедневные встречи и еженедельный доклад спонсору; сохранено правило ответа за два рабочих часа. **Процесс:** Подготовка и вовлечение участников. **Предшественники:** POC-01	**РП** Владелец продукта — обеспечивает участие специалистов продукта и подтверждает выделение их времени.
POC-06. Обучить команду и продемонстрировать работу с агентами Проведено обучение новому процессу: участникам представлены их действия, точки HITL и порядок обмена материалами. ИИ-инженер на задачах продукта продемонстрировал постановку и уточнение задачи, ревью, возврат замечаний и повторную проверку исправлений. Сохранены материалы, состав участников и ссылка на демонстрацию либо её протокол; вопросы разобраны. Демонстрация в ходе обычной работы PoC засчитывается, отдельное занятие не обязательно. Самостоятельная работа с агентами и обязательная практическая проба на PoC не требуются. При желании специалисты подключаются через инженера. **Процесс:** Подготовка и вовлечение участников. **Предшественники:** Начало PoC и доступные примеры задач и взаимодействий с агентами.	**РП** ИИ-инженер — готовит и проводит демонстрацию, разъясняет действия и разбирает вопросы; аналитик, разработчик и тестировщик — проходят обучение и знакомятся с действиями в своей роли.

Обеспечение технической готовности

Работа и результат	Ответственный и соисполнители
РОС-07. Уточнить инфраструктуру и доступы для Пилота Обновлена карточка инфраструктуры: инференс, ИИ-платформа, трекер, Git, CI/CD, анализ кода, артефакты, чат, документация и тестовые среды по применимости. Записаны размещение, связность, мощность для планового потока и права людей, систем и платформы. Отличия от временного контура РоС перечислены; при отсутствии отличий подтверждена применимость существующей конфигурации. Сохранены технические согласования и согласование ИБ. **Процесс:** Обеспечение технической готовности. **Предшественники:** РОС-01	**DevOps** ИИ-инженер — задаёт требования платформы и агентов; РП — организует согласования и предоставляет состав участников.; Архитектор — при необходимости проверяет изменившиеся границы компонентов и доступов при смене контура.
РОС-08. Подготовить системы и ресурсы Пилота По карточке инфраструктуры развёрнуты, перенастроены либо подключены необходимые системы. Проверены работа инференса, лимиты и доступная мощность для запланированной нагрузки. Существующие компоненты сохраняются, если удовлетворяют требованиям Пилота; результаты записаны в карточке инфраструктуры. **Процесс:** Обеспечение технической готовности. **Предшественники:** РОС-07	**DevOps** ИИ-инженер — проверяет функции платформы и вызовы моделей.
РОС-09. Настроить связность и выдать доступы По карточке инфраструктуры настроены сетевые связи, служебные аккаунты и права системам и ИИ-платформе; участникам предоставлены необходимые аккаунты и права. При смене контура обновлены адреса, интеграции и ссылки; полномочия временного РоС не переносятся автоматически. Результаты настройки сохранены в карточке инфраструктуры. **Процесс:** Обеспечение технической готовности. **Предшественники:** РОС-08.	**DevOps** ИИ-инженер — уточняет операции интеграций; РП — подтверждает список участников и нужные им права.
РОС-10. Подготовить тестовые среды для потока Пилота Проверены развёртывание выбранной ревизии, подготовка и сброс данных, получение логов и e2e-прогон. Результаты сохранены в карточке инфраструктуры. Среды воспроизводят необходимые условия продуктивной среды; отличия и ограничения перечислены. Для используемых зависимостей работают программно управляемые моки, эмуляторы и управление временем по требованиям выбранных задач. Агенты выполняют проверки в этих средах. **Процесс:** Обеспечение технической готовности. **Предшественники:** РОС-09.	**DevOps** Тестировщик — задаёт сценарии и подтверждает выполнение e2e; разработчик — обеспечивает сборку и запуск проверок.

Работа и результат	Ответственный и соисполнители
POC-11. Проверить техническую готовность из контура Пилота Из контура ИИ-платформы выполнены необходимые операции: чтение и обновление задачи, работа с репозиторием и MR, запуск CI, развёртывание артефакта по ID, тесты и получение результатов, отправка обращения в чат. Проверены только разрешённые операции; доказательства внесены в карточку инфраструктуры. Подтверждения временного PoC не заменяют проверки новых подключений. **Процесс:** Обеспечение технической готовности. **Предшественники:** POC-10.	**ИИ-инженер** DevOps — устраняет технические ошибки и повторяет зависимые проверки.
POC-12. Организовать личную проверку доступов участников Каждый участник вошёл в системы Пилота и выполнил необходимые ему действия: например, ответ в чате, просмотр задачи и MR, ревью, чтение отчёта о тестах. В карточке инфраструктуры сохранены отметки фактических проверок; проблемы доступа устранены. **Процесс:** Обеспечение технической готовности. **Предшественники:** POC-02, POC-09	**РП** ИИ-инженер, аналитик, разработчик, тестировщик и DevOps — проверяют операции в пределах назначенных им обязанностей.

Встраивание агентов в процессы разработки

Работа и результат	Ответственный и соисполнители
POC-13. Уточнить общий жизненный цикл задач и Git-процесс Обновлены статусная модель и Git-процесс для систем Пилота. Люди и агенты работают в общем процессе; описаны выбор части задач агентам, статус или поручение для запуска, связь задачи с веткой/MR, проверки, ревью и интеграция. В описаниях сопоставлены контакты людей и агентов на PoC и Пилоте: поручение, вопросы и ответы, HITL, ревью, возврат замечаний, приёмка и передача человеку. Для каждого предусмотренного взаимодействия указана проверка на PoC либо дополнительная проверка в POC-17. Смена трекера, Git или среды допустима, если сохраняется смысл взаимодействий и участие соответствующих ролей; упрощение не должно исключать эти взаимодействия из проверки. **Процесс:** Встраивание агентов в процессы разработки. **Предшественники:** POC-01	**РП** Аналитик — уточняет поступление требований; разработчик — ветки, MR и интеграцию; DevOps — автоматические действия конвейера.

Работа и результат	Ответственный и соисполнители
РОС-14. Описать точки участия человека В маршруте задачи записаны точки, следующие из task/Git-процессов, и обращения, которые агент формирует по навыкам при вопросе, ошибке или недостатке полномочий. Указаны необходимые роли проверяющих, точки HITL, материалы запроса и условия продолжения после ответа. Конкретные исполнители, их время и замещения берутся из назначений в паспорте, подтверждённых в РОС-02, без второго списка назначений. Запросы HITL направляются в чат с событием и материалами. **Процесс:** Встраивание агентов в процессы разработки. **Предшественники:** РОС-02, РОС-13	**РП** ИИ-инженер — описывает обращения по навыкам; аналитик, разработчик и тестировщик — подтверждают свои проверки и решения.
РОС-15. Определить проверки и приёмку результатов В правилах приёмки в ЖЦ задач указаны обязательные проверки по типам работ, сохраняемые доказательства, участники технической и продуктовой приёмки, оформление дефекта и повторной проверки. Автоматические проверки сохраняются при любом режиме участия человека. **Процесс:** Встраивание агентов в процессы разработки. **Предшественники:** РОС-13.	**Тестировщик** Разработчик — определяет проверки кода и MR; аналитик — проверку требований; владелец продукта — продуктовую приёмку.; Архитектор — при необходимости определяет проверки контрактов и совместимости для затронутых архитектурных решений.
РОС-16. Обеспечить передачу задачи от агента человеку В жизненном цикле задач описано, как по решению команды прекратить выполнение агентом и передать задачу человеку, сохранив изменения и материалы работы. Команда может забрать задачу в любой момент без обязательного объяснения причин; заранее определять перечень оснований, предел попыток и условия такого решения на РоС не требуется. Событие передачи сохраняется в рабочих системах для расчёта метрики на Пилоте. **Процесс:** Встраивание агентов в процессы разработки. **Предшественники:** РОС-13	**ИИ-инженер** РП — подтверждает порядок изменения исполнителя; разработчик — проверяет получение изменений и возможность продолжить работу.

Работа и результат	Ответственный и соисполнители
РОС-17. Проверить рабочий маршрут в контуре Пилота Сохранён сквозной прогон по согласованному процессу: запуск, выполнение, MR, проверки, запрос HITL, исправление замечания и приёмка. Проверена передача задачи человеку по решению команды: агент прекратил выполнение, изменения сохранены, человек может продолжить работу, событие передачи записано. Если передачи не было в обычной работе, выполнена учебная проба. Дополнительно проверены взаимодействия, которые отсутствовали или существенно отличались в процессе PoC. Результаты связаны с задачей и версиями правил. **Процесс:** Встраивание агентов в процессы разработки. **Предшественники:** РОС-12, РОС-14, РОС-15, РОС-16, РОС-20.	**ИИ-инженер** Аналитик, разработчик и тестировщик — выполняют проверки и решения по своим ролям; РП — устраняет организационные задержки.

Управление эффективностью работы команды и агентов

Работа и результат	Ответственный и соисполнители
РОС-18. Адаптировать агентов на задачах PoC По пробам разобраны ошибки, замечания и затруднения; доработаны инструкции, навыки, память и инструменты, затем повторены зависимые проверки. Сведения о продукте собираются по фактическим вопросам; отдельный индекс контекста не требуется. В журнале улучшений сохранены изменение, причина и результаты повторной проверки. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** Начало PoC по решению предыдущего гейта.	**ИИ-инженер** Аналитик — объясняет требования; разработчик — замечания к коду; тестировщик — ошибки тестов и проверки качества.; Архитектор — при необходимости уточняет архитектурные решения, которые нельзя восстановить только из кода.
РОС-19. Свести результаты проверок PoC В журнале задач, журнале улучшений и журнале показателей собраны ссылки на задачи, версии инструкций и навыков, MR, прогоны и замечания. Для проверенных видов работ записаны результат, необходимая помощь человека и ограничения. Указано, какие возможности проверены, а какие не проверялись. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** РОС-18	**ИИ-инженер** Аналитик — подтверждает соответствие требованиям; разработчик — результаты ревью; тестировщик — качество проверок и найденные дефекты.

Работа и результат	Ответственный и соисполнители
РОС-20. Подготовить и проверить рабочую версию агентов Для старта Пилота сохранены версии инструкций, навыков, контекста и настроек инструментов. После исправлений повторно проверены необходимые сценарии выбранной области; есть ссылки на результаты и восстановимую предыдущую версию. Новые адреса систем и ограничения Пилота учтены. Обязательных неисправленных замечаний нет. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** РОС-11.	**ИИ-инженер** Разработчик и тестировщик — подтверждают устранение замечаний в своих областях.
РОС-21. Согласовать метрики и правила мониторинга Заполнен, проверен и согласован план мониторинга: показатели результата, процесса и баланса, связь с целями, расчёты по описанию метрик, источники, периодичность и правила проверки. Записаны текущие значения и их происхождение, предварительные целевые ориентиры либо допустимые пределы, период наблюдения и условия итогового решения. На Пилоте расчёты и ориентиры уточняют по этапному порядку. Предусмотрены метрики ожиданий: ревью, тестирование и другие HITL, очереди агентов, зависимости, лимит параллелизма, среды и инфраструктурные отказы. Указаны события начала/конца, источники, открытые ожидания и ограничения данных; неприменимость отдельных причин обоснована. Включена метрика передачи задач от агентов людям: количество передач за период и доля переданных задач среди задач, взятых агентами в работу в той же когорте. На Пилоте её используют для разбора и уточнения условий передачи; объяснение причин не является обязательным условием самой передачи. РП согласует оценки объёма работ; трудозатраты людей на проверки, помощь и доработки включаются в учёт. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** РОС-13.	**РП** Владелец продукта — согласует связь с целями; ИИ-инженер — проверяет доступность и ограничения данных.

Работа и результат	Ответственный и соисполнители
РОС-22. Рассчитать и согласовать с РП оценочные значения для сравнения По плану мониторинга рассчитаны оценочные значения для выбранной области: объём работ в идеальных часах и человеческие затраты, используемые при сравнении. Для каждой оценки сохранены состав работ, основания и допущения; оценки отделены от зарегистрированных фактов. РП проверил оценки и подтвердил их либо внёс уточнения; сохранены согласованные значения и запись согласования. При недостатке данных ограничения указаны явно; отсутствие значения не заменено нулём и не представлено как доказанный эффект. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** РОС-21	**ИИ-инженер** РП — проверяет, уточняет и согласует оценки; владелец продукта — подтверждает состав работ и область сравнения.
РОС-23. Настроить и проверить ежедневный сбор сведений Агент по описаниям процессов и плану мониторинга собирает события из доступных рабочих систем, связывает их с задачами, рассчитывает объём и время и вычисляет показатели. Выполнен пробный сбор за день, повторный сбор и сбор заданного периода без дублирования. В плане мониторинга указан ежедневный автозапуск либо запрос агенту собрать прошедший день/заданные дни; ошибки сбора и неполнота данных отражаются явно. Проверено получение события передачи задачи от агента человеку для метрики Пилота, а также доступность событий применимых ожиданий из плана мониторинга; пробелы отмечены явно. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** РОС-09, РОС-21.	**ИИ-инженер** DevOps — обеспечивает технические доступы к источникам; РП — подтверждает расписание и способ запуска.
РОС-24. Проверить пробную недельную сводку Агент подготовил недельную сводку по имеющимся данным PoC с указанием периода и контура. Проверены связи с первичными событиями, полнота, объём и человеческие затраты. Выбранные расхождения разобраны, обязательные ошибки сбора и расчёта устранены; назначена дата первого разбора Пилота. Данные временного PoC не выдаются за исходную базу Пилота без проверки сопоставимости. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** РОС-22, РОС-23.	**РП** ИИ-инженер — исправляет сбор и расчёты; владелец продукта — проверяет представление результатов и выводы для целей внедрения.

Чек-лист завершения Пилота

GATE-03 · Продуктовая команда · Завершение Пилота.

Гейт проверяет достижение согласованных показателей и закрепление рабочего способа совместной работы команды с агентами. Все перечисленные ниже работы выполняются на Пилоте.

Если согласованные целевые показатели достигнуты и обязательные условия чек-листа выполнены, РП фиксирует успешное завершение Пилота. Дополнительное решение Спонсора о принятии достигнутого результата не требуется. При недостижении показателей или невыполнении других обязательных условий Спонсор решает продолжить Пилот, изменить его условия либо остановить. После изменения целей их достижение проверяется при следующем подведении итогов.

Успешное завершение Пилота команды может учитываться на уровне предприятия, но само по себе не запускает масштабирование. Решения и работы предприятия в этот документ не входят.

Работы Пилота · Предыдущий гейт — готовность к Пилоту.

Условия успешного завершения

- [ ] **PF-01. Данные для расчёта показателей собраны и подтверждены.** По плану мониторинга подготовлены наборы данных («витрины») для расчёта показателей. Первичные сведения о задачах, действиях людей и агентов, проверках и передачах связаны с записями трекера, Git, чата и журналов; происхождение сведений и внесённые уточнения прослеживаются. Рассчитанные на их основе оценки объёма работ в идеальных часах и затрат времени людей сохранены вместе с основаниями и подтверждены РП. В витринах факты отделены от оценок, отражены пропуски и незавершённые задачи, учтены помощь агентам, ревью и исправления. В наборах данных отражены применявшиеся конфигурации инфраструктуры и инференса и периоды их использования. Сохранены версии данных и правила расчёта: повторный расчёт по той же версии даёт те же значения показателей, а повторный сбор не создаёт дубли. Работы: PIL-14, PIL-15, PIL-16.
- [ ] **PF-02. Согласованные целевые показатели Пилота достигнуты.** В плане мониторинга задан полный набор показателей успешности Пилота: результат, а также выбранные показатели процесса и баланса, включая качество, нагрузку людей и стоимость по целям проекта. В итогах Пилота приведены целевые и фактические значения, период, область и ссылки на расчёты по подтверждённым наборам данных. Для каждого показателя выполнены согласованные пороги или допустимые пределы; использованы данные и правила расчёта из PF-01. Прогноз или план улучшений не считается достигнутым результатом. Работы: PIL-01, PIL-19, PIL-20, PIL-22, PIL-03.
- [ ] **PF-03. Рабочие процессы и инструкции актуализированы по результатам Пилота.** Жизненный цикл задач, Git-процесс и взаимодействие с агентами описывают фактическую работу: запуск, HITL, приёмку и передачу человеку. Сохранены ссылки на действующие версии процессов и инструкций и примеры выполненных задач. Работы: PIL-02, PIL-05, PIL-10, PIL-11, PIL-12, PIL-18; при изменении контроля — PIL-13.
- [ ] **PF-04. Обеспечено дальнейшее сопровождение.** В разделе «Команда проекта» паспорта назначены конкретные участники, подтверждены их время, доступы и замещения для поддержки платформы, сред, мониторинга и развития агентов. Назначен ИИ-инженер с доступами к инструкциям, навыкам и средствам диагностики. Указано, куда поступают замечания и кто их

разбирает. Для выполненных улучшений в журнале улучшений указаны фактические исполнители и их команды; при участии ИИ-инженера внедрения записано, почему работу не выполнял инженер продукта. Полная передача сопровождения специалистам продукта не обязательна. Работа: PIL-06; при выявленной необходимости улучшений — PIL-17.

- [ ] **PF-05. Требования к инфраструктуре и инференсу актуализированы, изменения учтены в расчётах.** В карточке инфраструктуры уточнены требования по результатам Пилота и описана итоговая конфигурация: размещение систем, мощности, модели и лимиты инференса, связность, доступы и тестовые среды. Изменения относительно первоначальных предположений сохранены с датами и основаниями. Итоговые показатели связаны с этой конфигурацией; затраты рассчитаны с учётом необходимых ей ресурсов и фактических условий использования инференса. Если конфигурация менялась в период наблюдения, в данных различаются периоды её использования, а результаты на прежней конфигурации не выдаются за результаты на итоговой. Работы: PIL-07, PIL-08, PIL-09, PIL-19, PIL-20.

- [ ] **PF-06. Оформлены итоги и ограничения применения результатов Пилота.** В итогах Пилота перечислены типы задач и участки процесса, рабочие версии агентов, режимы проверки, необходимые средства и участие людей, ограничения и исключения. Есть ссылки на проверенные инструкции и примеры. Отмечено, в каких условиях получены результаты и на какие задачи распространяется итоговый вывод. Пакет результатов описывает выполненный Пилот; следующие команды и волны не планируются. Работа: PIL-21.

Подведение итогов

РП готовит итоги и фиксирует статус Пилота — PIL-03. При выполнении PF-01–06 Пилот фиксируется как **успешно завершённый**; отдельного разрешения Спонсора не требуется.

Если показатели не достигнуты или другие обязательные условия не выполнены, Спонсор определяет дальнейшие действия — PIL-04: **продолжить Пилот / изменить условия и продолжить Пилот / остановить Пилот без успешного завершения**.

В итогах сохраняются:

- Статус Пилота и дата подведения итогов: …
- Целевые и фактические показатели, версия согласованных условий и ссылки на подтверждения: …
- Выполнение PF-01–06 и незавершённые работы: …
- Проверенная область применения, режимы и ограничения: …
- Дальнейшее сопровождение и ответственные: …
- При продолжении — решение Спонсора, изменения условий, работы и дата следующего подведения итогов: …
- При остановке — решение Спонсора и порядок обращения с текущими задачами и результатами: …

Изменение целей, периода наблюдения или области фиксируется с основаниями и согласованиями. Пилот остаётся продолжающимся до проверки выполнения новых условий; пересмотр цели сам по себе не превращает прежний результат в успешное завершение. Фактические значения и история изменений сохраняются.

Работы на этапе Пилота

Работы распределены по пяти процессам. У каждой один ответственный; в проекте назначается конкретный человек. Соисполнители указаны вместе с вкладом. Ссылки на результаты сохраняют в существующем трекере, Git, карточках и журналах; отдельный ручной учёт каждого действия не нужен.

Команда ведёт текущие задачи с агентами, ИИ-инженер обеспечивает сопровождение и развитие. Агент ежедневно собирает сведения, РП организует недельный разбор показателей и оценок, специалисты проверяют результаты и предлагают улучшения. Выявленные ограничения устраняют и повторно проверяют. На накопленных данных оценивают результат, побочные последствия и достаточность сопровождения, фиксируют границы подтверждённого применения и проводят итоговую встречу. Период наблюдения и ориентиры задаются в проекте. Если они пересматриваются при продолжении Пилота, сохраняются основания, согласования и дата следующей проверки; методология не назначает универсальный процент эффекта или обязательное число недель.

В предшественниках указаны непосредственные зависимости. Работы, уже связанные через других предшественников, повторно не перечисляются; несколько ссылок обозначают независимые ветви подготовки результата. Условная работа учитывается только при её применимости. Предшественники относятся к завершению работы и не запрещают начать независимые части раньше. Регулярные работы ведутся параллельно весь Пилот; на гейте рассматривают их результаты за согласованный период. Для подведения итогов нужны статусы всех работ, включая незавершённые. Работа PIL-04 выполняется только при недостижении показателей или невыполнении других обязательных условий; для успешного завершения её выполнение не требуется. PIL-13 выполняется при изменении режима контроля, PIL-17 — при необходимости улучшений. Если соответствующей необходимости не возникло, работа отмечается как неприменимая с кратким основанием. Условная работа в предшественниках не требует искусственно создавать изменение или проблему.

Управление внедрением

Работа и результат	Ответственный и соисполнители
PIL-01. Уточнить критерии и период итоговой оценки В плане мониторинга проверены связь с целями, область, полный набор целевых показателей результата, процесса и баланса, их пороги и допустимые пределы, период наблюдения, достаточность данных и правила сравнения. Согласованные показатели и условия служат основанием для фиксации успешного завершения. В ходе Пилота уточняют процедуру вычисления, особенно оценку объёма задач, и при необходимости предварительные целевые значения по этапному порядку. Сохраняют основания, прежние и новые значения, сопоставимые расчёты, версию и дату следующей проверки. Изменения целей и условий успешности согласуются участниками и утверждаются Спонсором по принятому порядку. **Процесс:** Управление внедрением. **Предшественники:** Начало Пилота по решению предыдущего гейта.	**РП** Владелец продукта — подтверждает цели и область; ИИ-инженер — проверяет измеримость; Спонсор — утверждает изменения целей и условий успешности по принятому порядку.

Работа и результат	Ответственный и соисполнители
PIL-02. Проводить ежедневный и недельный разбор внедрения На ежедневных встречах разобраны вопросы без решения, у действий есть ответственные и сроки. Еженедельно рассмотрены результаты общего разбора эффективности, выполнение работ, ресурсы и препятствия. Спонсору представлены изменения и нерешённые вопросы; независимый повторный анализ тех же метрик не проводится. В недельных сводках сохранены решения и выполнение поручений. **Процесс:** Управление внедрением. **Предшественники:** Начало Пилота по решению предыдущего гейта.	**РП** Владелец продукта — разрешает продуктовые приоритеты; ИИ-инженер и специалисты продукта — представляют результаты и выполняют принятые действия.
PIL-03. Подвести итоги и зафиксировать статус Пилота В итогах Пилота собраны статусы PF-01–06, целевые и фактические показатели, подтверждения и ограничения. При выполнении согласованных показателей и остальных обязательных условий РП фиксирует успешное завершение и сообщает результат участникам. Если условия не выполнены, РП представляет Спонсору отклонения и предложение о продолжении, изменении условий или остановке; сохраняет принятое решение. Подготовка материалов не требует предварительного закрытия всех пунктов. **Процесс:** Управление внедрением. **Предшественники:** PIL-02, PIL-21, PIL-22.	**РП** Владелец продукта — подтверждает продуктовый результат; ИИ-инженер — расчёты, технические результаты и ограничения.
PIL-04. Определить дальнейшие действия при невыполнении условий Пилота Работа выполняется, если целевые показатели не достигнуты или другие обязательные условия не выполнены. Спонсор выбирает продолжение Пилота, изменение условий с продолжением либо остановку без успешного завершения. При изменении условий сохранены основания, согласования участников, новая версия целей и дата следующей проверки. При достигнутых показателях и выполненных остальных условиях отдельное решение Спонсора не требуется. **Процесс:** Управление внедрением. **Предшественники:** PIL-03; невыполнение условий успешного завершения.	**Спонсор** РП — готовит варианты дальнейших действий, сохраняет решение и отражает работы и сроки в плане.

Подготовка и вовлечение участников

Работа и результат	Ответственный и соисполнители
PIL-05. Обеспечить освоение обязанностей участниками Пилота По реальным задачам Пилота подтверждено, какие действия специалисты продукта выполняют непосредственно: поручение и уточнение задач, ответы, ревью, проверки и приёмка. Подтверждения связаны с участниками и их ролями в трекере, Git и чате. При затруднениях обеспечены помощь, дообучение, время или замещение; их результат подтверждён дальнейшими действиями. Фактическая помощь ИИ-инженера видна в данных, выполнение им всех действий не выдаётся за освоение командой. Искусственное упражнение для отчёта не требуется. Работоспособность и актуальность самого маршрута подтверждаются в PIL-10. **Процесс:** Подготовка и вовлечение участников. **Предшественники:** Начало Пилота по решению предыдущего гейта.	**РП** Владелец продукта — обеспечивает поток и приоритет; аналитик — требования и ответы; разработчик — ревью и интеграцию; тестировщик — проверку качества; ИИ-инженер — помощь при затруднениях.
PIL-06. Закрепить дальнейшее сопровождение и развитие агентов В разделе «Команда проекта» паспорта обновлены роли, конкретные участники, выделенное время, замещения и порядок обращения за помощью. Записи журнала улучшений показывают фактически выполненные изменения, исполнителей и их команды. Если работу выполнял ИИ-инженер внедрения, пояснено, почему её не выполнял инженер продукта. Сохранённое инженерное сопровождение согласовано по объёму и ресурсу; полная передача не требуется. Назначен ИИ-инженер с доступами к инструкциям, навыкам и диагностике и временем на разбор замечаний. Записано общее место сохранения замечаний и способ их передачи ему. Если улучшения потребовались, работа по ним даёт практику; при отсутствии необходимости отдельное упражнение не проводится. **Процесс:** Подготовка и вовлечение участников. **Предшественники:** PIL-05	**РП** Владелец продукта — подтверждает участие специалистов; ИИ-инженер — объём инженерной помощи; DevOps — поддержку платформы и сред.

Обеспечение технической готовности

Работа и результат	Ответственный и соисполнители
PIL-07. Поддерживать рабочую инфраструктуру и доступы Изменения конфигурации, лимитов, подключений и прав внесены в карточку инфраструктуры с датами применения и основаниями; сохранены предыдущие значения. После затрагивающих поток изменений выполнены зависимые проверки из рабочего контура. Сбои и действия по их устранению связаны с рабочими задачами; накоплены данные доступности и использования ресурсов. **Процесс:** Обеспечение технической готовности. **Предшественники:** Начало Пилота по решению предыдущего гейта.	**DevOps** ИИ-инженер — проверяет влияние на платформу и инструменты агентов.
PIL-08. Обеспечивать воспроизводимость тестирования В потоке Пилота выполняются сборка и развёртывание по точной ревизии, подготовка данных и e2e-проверки. Сохранены версии, результаты и диагностика прогонов. При выявленных расхождениях среды с продуктивной доработаны данные, управляемые моки, эмуляторы и другие необходимые средства. Конфигурация и результаты проверок отражены в карточке инфраструктуры. **Процесс:** Обеспечение технической готовности. **Предшественники:** PIL-07	**DevOps** Тестировщик — проверяет пригодность сценариев и данных; разработчик — воспроизводимость сборки и развёртывания.
PIL-09. Актуализировать требования к инфраструктуре и инференсу по результатам Пилота В карточке инфраструктуры уточнены требования к ресурсам, параллелизму, моделям и лимитам инференса, размещению, связности, доступам и средам. Записана итоговая используемая конфигурация, изменения относительно исходных предположений, даты и основания. Для расчётов предоставлены необходимые ресурсы, условия использования и стоимость; указано, в какие периоды применялась каждая конфигурация. Сохранены согласования уточнённых требований. Подготовка инфраструктуры следующих команд и волн не входит в эту работу. **Процесс:** Обеспечение технической готовности. **Предшественники:** PIL-08.	**DevOps** ИИ-инженер — уточняет требования агентов и инференса, связывает конфигурации с данными наблюдений; РП — организует согласование ресурсов и стоимости.; Архитектор — при необходимости проверяет влияние итоговой конфигурации на границы компонентов и зависимости.

Встраивание агентов в процессы разработки

Работа и результат	Ответственный и соисполнители
PIL-10. Вести общий процесс задач и Git Описания процессов актуализируются по фактической работе команды и агентов. В них закреплены обязанности ролей, точки HITL, приёмка и передача задач; инструкции и решения доступны в общих системах. Команда и агенты используют согласованные жизненный цикл задач и Git-процесс. Сохранены связи задач с MR, проверками и приёмкой; изменения правил отражены в их описаниях. Нестандартные маршруты и ручные обходы разобраны, необходимые исправления проверены. **Процесс:** Встраивание агентов в процессы разработки. **Предшественники:** Начало Пилота по решению предыдущего гейта.	**РП** Аналитик — обеспечивает поступление требований; разработчик — ревью и интеграцию; тестировщик — результаты проверок.
PIL-11. Обеспечивать передачу задач от агентов людям При решении команды забрать задачу агент прекращает выполнение, сохраняются изменения, нужная ревизия, контекст и результаты проверок. Человек получает доступ к сохранённой работе и может продолжить её; событие передачи и фактическое продолжение связаны с исходной задачей в журнале задач. Технические ошибки передачи устранены либо явно ограничивают её применение. Если передач не было, отмечено отсутствие событий; искусственная передача для отчёта не нужна. Объяснение причины не является обязательным условием. Частота, доля и затраты передач вычисляются в PIL-14–16, меры выбираются в PIL-17; здесь подтверждается корректность самой передачи. **Процесс:** Встраивание агентов в процессы разработки. **Предшественники:** PIL-10	**РП** ИИ-инженер — обеспечивает фиксацию передачи и разбирает технические затруднения; разработчик — продолжает переданные ему задачи.
PIL-12. Поддерживать работоспособность обращений HITL Обращения направляются по согласованным событиям нужной роли, содержат материалы и ревизию; ответ и результат проверки доступны агенту и правильно продолжают задачу. Выявленные ошибки формирования, доставки, чтения ответа и переходов устранены, затронутые цепочки повторно проверены. Изменения отражены в ЖЦ задач и правилах Git. При отсутствии ошибок сохранены подтверждения текущей работы, дополнительные изменения не требуются. Длительность ожиданий рассчитывается в PIL-14–16, меры по потерям выбираются в PIL-17; нехватка времени, замещения или навыка устраняется в PIL-05. **Процесс:** Встраивание агентов в процессы разработки. **Предшественники:** PIL-10	**РП** Аналитик, разработчик и тестировщик — проверяют получение обращений, пригодность материалов и использование своих ответов; ИИ-инженер — исправляет обращения, которые не доходят до нужного участника.

Управление эффективностью работы команды и агентов

Работа и результат	Ответственный и соисполнители
PIL-14. Ежедневно собирать сведения и вычислять метрики Агент по плану мониторинга ежедневно собирает события из трекера, Git, чата, журналов агентов и тестовых сред. Сохранены связанные с задачами первичные сведения, ссылки на записи источников и внесённые уточнения; пропущенные дни собраны по запросу за заданный период. Повторный сбор не дублирует события. Ошибки и неполнота видны в сводках и исправлены либо учтены как ограничения. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** PIL-01	**ИИ-инженер** DevOps — восстанавливает техническую доступность источников; РП — обеспечивает ежедневный запрос агенту, если автозапуск не используется.
PIL-15. Согласовывать оценочные значения с РП По плану мониторинга агент рассчитывает объём работ в идеальных часах и оценочные человеческие затраты. У оценок сохранены состав работ, основания, допущения и версия; РП проверяет и подтверждает их либо уточняет. Корректировки после решения задач сохраняются с пояснением изменения. По накопленным случаям ИИ-инженер совершенствует процедуру оценки объёма; версия правил и затронутые наборы данных обновляются с сохранением сопоставимости. Согласованные оценки включены в подготовленные наборы данных для расчёта показателей со ссылками на основания и подтверждение РП. Факты и оценки различаются, участие людей в проверках и исправлениях не пропускается. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** PIL-14	**ИИ-инженер** РП — проверяет, уточняет и согласует оценки; владелец продукта — подтверждает существенные изменения состава работ.

Работа и результат	Ответственный и соисполнители
PIL-16. Проверять недельные сводки и полноту данных Агент готовит недельные сводки; РП с командой проверяет существенные расхождения и выбранные аномалии. Сохранены период, область, состав задач, согласованные оценки и ссылки на первичные события. Учтены незавершённые задачи, повторные попытки, помощь людей и передачи; отсутствие данных не заменено нулевыми затратами. Проверены подготовленные наборы данных («витрины»), связывающие первичные сведения и подтверждённые оценки с расчётом показателей. Сохранены версии наборов данных и правил; повторный расчёт по тем же данным даёт те же значения. Витрина может быть автоматически сформированной таблицей или выгрузкой из рабочих систем, отдельного ручного реестра не требуется. Ошибки исправлены с пересчётом затронутых периодов. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** PIL-15.	**РП** ИИ-инженер — исправляет сбор и расчёты; специалисты продукта — уточняют проверяемые расхождения.
PIL-17. Улучшать агентов и способ работы по выявленным ограничениям Работа выполняется при выявленной необходимости улучшений. Здесь выбирают меры по причинам потерь и проверяют эффект; изменения маршрута выполняются в PIL-10–12, меры по людям — в PIL-05, технические изменения — в PIL-07/08. Специализированная проба смены контроля ведётся в PIL-13; одно изменение и его проверка повторно не заводятся. Из метрик и замечаний выбраны ограничения; для каждого улучшения записаны ожидаемый результат, область пробы и ответственный. Улучшены инструкции, навыки, контекст, инструменты либо процессы и среды. В журнале улучшений сохранены изменения, повторные проверки и решение сохранить, доработать или отменить изменение. Организационные проблемы не подменяются доработкой навыков. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** PIL-14	**ИИ-инженер** РП — обеспечивает организационные изменения; аналитик — знания о требованиях; разработчик и тестировщик — замечания и проверки; DevOps — изменения сред и инструментов.; Архитектор — при необходимости разбирает архитектурные причины повторяющихся затруднений при их наличии.

Работа и результат	Ответственный и соисполнители
PIL-13. Проверять изменения режима человеческого контроля Работа выполняется, если на Пилоте выбрана проба изменения режима для отдельной области. Это специализированный цикл проверки гипотезы в процессе эффективности; одна проба не дублируется отдельной работой в PIL-17. До пробы записаны гипотеза, метрики, область и условия возврата. После ограниченной пробы сопоставлены результат, возвраты с тестирования, поздние дефекты и затраты людей на исправления; сохранено решение оставить режим, доработать либо вернуться к HITL. Применённые изменения правил отражены в PIL-10, а корректность обмена — в PIL-12. Для HOTL проверена регистрация и исправление асинхронных замечаний. При сохранении HITL в итогах Пилота фиксируется, что режим не менялся. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** PIL-10, PIL-14.	**РП** ИИ-инженер — настраивает пробу и сбор сведений; разработчик и тестировщик — подтверждают достаточность контроля; владелец продукта — согласует область применения.; Архитектор — при необходимости уточняет архитектурные риски и сохраняемые проверки при изменении контроля.
PIL-18. Закрепить проверенные версии и возможность восстановления В Git и настройках платформы сохранены используемые рабочие версии инструкций, навыков, контекста и инструментов; участники имеют доступ к материалам в своей зоне ответственности. Если версии менялись, сохранены результаты повторных проверок и способ восстановления предыдущей рабочей версии; изменения связаны с периодами наблюдения. При отсутствии изменений сохраняется версия, с которой команда работала на Пилоте. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** Начало Пилота; при выполнении улучшений — PIL-17.	**ИИ-инженер** Разработчик и тестировщик — подтверждают результаты повторных проверок в своей области.

Работа и результат	Ответственный и соисполнители
PIL-19. Подвести количественные итоги по целям внедрения Из подготовленных наборов данных по сохранённым правилам вычислены показатели. В итогах Пилота сопоставлены показатели результата с согласованными ориентирами и базой сравнения; приведены ссылки на использованные версии данных и расчётов. Указаны период, состав работ, объём наблюдений, фактические и оценочные величины, ограничения сопоставления. Результаты связаны с итоговой конфигурацией из карточки инфраструктуры; периоды работы на прежних конфигурациях выделены. При недостатке наблюдений на итоговой конфигурации это отражено в выводе; её результат не подменяется пересчётом фактических сроков на предположениях. Экономия в производстве ПО показана отдельно от затрат на внедрение и сопровождение. Для каждого согласованного целевого показателя записано, достигнут он или нет. При недостижении описаны отклонения и их причины для решения о продолжении, изменении условий либо остановке. Факты, оценки и прогнозы показаны раздельно. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** PIL-09, PIL-16, PIL-18.	**РП** ИИ-инженер — готовит расчёты и подтверждения; владелец продукта — подтверждает состав работ и продуктовый результат.
PIL-20. Проверить качество, нагрузку людей и стоимость По плану мониторинга рассмотрены дефекты и возвраты, ожидание HITL, передачи людям, затраты проверяющих и доработок, инференса, инфраструктуры и сопровождения. Результаты сопоставлены с согласованными пределами; отклонения, включая последствия HOTL при его применении, перечислены и связаны с действиями. Затраты на инфраструктуру и инференс рассчитаны по итоговой конфигурации из карточки инфраструктуры, включая необходимые ей ресурсы и действующие условия использования. Фактические расходы за весь Пилот, включая прежние конфигурации, показаны отдельно от оценки затрат дальнейшей работы. Снижение ручных проверок не скрывает рост последующих исправлений. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** PIL-09, PIL-11, PIL-13, PIL-16	**РП** Тестировщик — подтверждает дефекты и возвраты; разработчик — доработки; ИИ-инженер — расчёты человеческих затрат и инференса; DevOps — расходы и ограничения инфраструктуры.

Работа и результат	Ответственный и соисполнители
PIL-21. Описать проверенный способ работы и границы его применения В итогах Пилота собраны ссылки на действующие процессы, версии агентов, инструкции, примеры задач и сводки. Указаны применимые типы задач, необходимые проверки, условия инфраструктуры, роль специалистов и сопровождения, ограничения и исключения. Зафиксированы условия, в которых получены результаты, и ограничения итогового вывода. Материалы оформлены для дальнейшего использования командой; внедрение в другие команды здесь не планируется. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** PIL-06, PIL-12, PIL-19, PIL-20.	**ИИ-инженер** РП — подтверждает организационные условия; владелец продукта — границы продуктового применения; разработчик и тестировщик — ограничения проверок.; Архитектор — при необходимости подтверждает архитектурные ограничения применимости способа работы.
PIL-22. Устранить блокирующие отклонения и подтвердить результат Выявленные при итоговой оценке нарушения критериев связаны с работами в трекере. Для устранённых нарушений сохранены повторные проверки и обновлённые показатели за достаточный период наблюдения. Неустранённые нарушения и их влияние перечислены в итогах Пилота. При изменении условий Пилота сохранено решение о продолжении и следующей проверке; само решение не считается устранением нарушения. При отсутствии нарушений сохранён вывод проверки. **Процесс:** Управление эффективностью работы команды и агентов. **Предшественники:** PIL-19, PIL-20	**РП** ИИ-инженер — исправляет агентов и расчёты; разработчик и тестировщик — проверяют исправления; DevOps — устраняет инфраструктурные причины.

Чек-лист развития условий команды для работы с агентами

Внедрение в продуктовые команды. Форма примера для кабинета «Меридиан Сервис».

Чек-лист помогает оценить, насколько процессы и средства продукта поддерживают автономную работу. **Это не блокирующий чек-лист начала внедрения:** незакрытые пункты показывают, что развивать параллельно и какие виды работ пока оставлять людям или выполнять с HITL. Он дополняет обследование и план развития, а не заменяет готовность к PoC или готовность к Пилоту.

Шаблон

Продукт и паспорт: … . **Этап, дата оценки и версия:** … . **Область задач и используемые системы:** … .

Для каждого применимого пункта укажите «есть», «частично», «нет» или «неприменимо» с основанием и ссылкой на подтверждение. Отметка чек-бокса означает подтверждённое выполнение в названной области. При пробеле добавьте ссылку на работу развития либо временное ограничение. Ответственный и срок назначаются у работы в трекере, а не у каждого признака готовности. При повторной оценке сохраняются изменения состояния и подтверждений.

Итоговый балл и закрытие всего списка не требуются. Условия для будущей области не объявляются выполненными по проверке одной ограниченной задачи.

1. Постановка и движение задач

- [ ] Для выбранного потока есть источник требований, ожидаемый результат, границы и критерии приёмки. Большой запрос сначала уточняется и декомпозируется.
- [ ] Статусы и переходы трекера отражают реальную работу; понятны решения о начале, готовности, приёмке, передаче и отмене.
- [ ] В Git определены база, рабочие ветки, MR, ревью кода и тестов, повторная проверка и условия интеграции. Задача связана с нужными MR и ревизиями.
- [ ] Определён запуск агентом: разрешение в чате и/или статус трекера с точными условиями. Повторное поручение не создаёт второе выполнение.
- [ ] Для выбранных работ описаны HITL, HOTL или отсутствие плановой ручной проверки; роли и полномочия зафиксированы в процессах и паспорте.

Основные документы — процесс задач, Git-процесс и маршрут агента.

2. Требования и проверяемость результата

- [ ] Требования позволяют однозначно проверить изменяемое поведение; существенные развилки решены до реализации.
- [ ] Для выбранных задач названы основные, ошибочные и граничные сценарии, роли пользователей и нужные данные.

- [ ] Принятые решения и изменения требований сохраняются с версиями и источниками; код, проверка и приёмка относятся к понятной ревизии.
- [ ] Известные пробелы требований и проверок не скрываются; для них выбран временный способ работы или задача развития.

Требования уточняются по описанию работы аналитика; полный документ продукта не требуется до первой ограниченной задачи.

3. Сборка, данные и DEV/UAT-процесс

- [ ] Изменение по задаче можно собрать и развернуть на DEV без предварительного включения в общую ветку; сохраняются SHA и ID артефакта.
- [ ] Агент может выполнить сквозной E2E-сценарий выбранного потока на среде, похожей на production по существенным компонентам и взаимодействиям. Различия известны.
- [ ] Для параллельных задач изолированы ревизии, данные, моки, время и фоновые операции. Устаревший pipeline не заменяет новое развёртывание.
- [ ] Есть порядок резервирования, подготовки, очистки и возврата среды; действия доступны программно либо для остатка явно назначен HITL.
- [ ] Начальное состояние и тестовые данные воспроизводимы; сброс не затрагивает соседние задачи, а версии схемы и приложения совместимы.
- [ ] На UAT известны ревизия и условия проверки; они сохраняются во время приёмки. Агент имеет доступ к диагностике и может воспроизвести замечание.
- [ ] Изменения БД версионируются и проверяются. При отсутствии автоматических миграций есть воспроизводимая инструкция, исполнитель, подтверждение и восстановление при ошибке.

Развитие автоматических миграций и других инструментов идёт параллельно внедрению. До конкретной задачи с изменением БД обеспечивается достаточный рабочий способ. Гайд инфраструктуры.

4. Управляемые зависимости и выполнение сценариев

- [ ] Для каждой внешней зависимости выбран способ проверки: реальный тестовый контур, управляемый мок или сочетание; границы выводов известны.
- [ ] Моки управляются программно и позволяют получить нужные ответы, ошибки, задержки и журнал вызовов. Для значимых сценариев поддерживаются повторы и асинхронные события.
- [ ] Контекст моков изолирован по среде: изменение сценария или сброс истории не меняет поведение соседних задач.
- [ ] Для таймеров, расписаний и истечения сроков есть способ пройти сценарий без ожидания реальных часов и дней: виртуальное время или подходящее управление заданиями.
- [ ] Время, часовой пояс, локаль и глобальные настройки задаются профилем прогона и не меняют соседние проверки; для общих настроек без надёжного сброса используется отдельный процесс. Часы связанных компонентов согласованы, проверяется реальный механизм срабатывания.
- [ ] Оборудование заменено программно управляемым эмулятором с профилем, событиями, отказами, журналом и сбросом; агент не обращается к физическому устройству.
- [ ] Побочные эффекты направлены в тестовые приёмники; можно проверить фактические операции, дубли и отсутствие запрещённых эффектов.

- [ ] Случайные входы управляются seed или заданной последовательностью; неуспешные варианты сохраняются для повтора.
- [ ] Для асинхронности и гонок доступны управляемые события, барьеры/точки остановки и наблюдаемое условие завершения с ограниченным ожиданием.
- [ ] Удержанное состояние можно подготовить, прочитать, восстановить и сбросить; следующий прогон не наследует его от предыдущего.

Применимость определяется задачами продукта. Соответствующий механизм нужен до зависящего от него сценария; отсутствующие возможности включаются в план, а область автономной работы ограничивается проверяемыми задачами. Физическое оборудование при необходимости проверяет человек отдельно. Полный перечень механизмов. Моки, эмуляторы и эффекты · Время и контекст.

5. Тесты, доказательства и диагностика

- [ ] Существующие проверки инвентаризированы; известны команды, зависимости, способы подготовки данных и ограничения.
- [ ] Для изменяемой области выбран достаточный набор unit-, API-, UI- и сквозных проверок по применимости; отсутствующие виды отражены в плане.
- [ ] Повторяющиеся проверки можно воспроизводимо запускать агентом; результаты, ревизия, среда и данные сохраняются.
- [ ] Доказательства показывают проверенное поведение и ожидаемый результат, а не только наличие файлов тестов или зелёный статус запуска.
- [ ] Доступны журналы приложения, тестов, интеграций и агентов; их можно связать с задачей, средой и ревизией для разбора ошибки.
- [ ] До сокращения ручных проверок подтверждены автоматические замены, покрытие нужных сценариев и достаточность доказательств.

Полное покрытие всех видов не является условием начала внедрения. Достаточный способ проверки нужен для каждой выполняемой задачи. Проверка готовности тестовой инфраструктуры · Доказательства.

6. Участие людей и улучшение работы

- [ ] Назначены постановщик, проверяющие и принимающие в пределах их ответственности; у них есть время и доступ к материалам.
- [ ] HITL-запросы поступают через чат с событием и ссылками; ответы и последующие действия сохраняются. Напоминания не обнуляют ожидание.
- [ ] Задачу можно передать человеку с остановкой запусков агента, сохранением изменений и фиксацией нового исполнителя. Передача подтверждается фактом принятия; обязательного объяснения причины не требуется.
- [ ] Для HOTL назначены проверяющий и правило просмотра; находки имеют исполнителя и приоритет, существенная проблема может остановить затронутый поток.
- [ ] Для затруднений сохраняются факты; необходимые улучшения навыков и инструментов получают версию и проверяются на задачах. Неоднозначные наблюдения можно накапливать до решения об изменении.

Работа ИИ-инженера.

7. Метрики и обратная связь

Этот раздел применяется с PoC. На Подготовке в паспорте достаточно содержательной цели; выбор метрик и численных ориентиров ещё не требуется. На PoC проверяют доступные метрики, текущие значения и предварительные ориентиры, на Пилоте уточняют процедуру и при необходимости ориентиры по этапному порядку.

- [ ] В плане мониторинга заданы цель и срок проверки, метрики результата, процесса и баланса, границы трудозатрат, типы задач, исходная база и источники.
- [ ] Можно связать допуск задачи, версии требований, MR, проверки, запросы людям и запуск агента; пропуски видны отдельно. Для ожиданий HITL, очередей агентов, зависимостей, сред и отказов есть события начала/конца, открытые интервалы и их возраст.
- [ ] Агент собирает события раз в сутки по расписанию или просьбой, поддерживает сбор за пропущенные даты и пересборку без дублей.
- [ ] Идеальные часы и участие людей оцениваются в одинаковых границах; версии оценок, основания, подтверждения РП и причины изменений сохраняются.
- [ ] Показатели просматриваются во времени с отметками изменений и наблюдениями участников. Недельный разбор приводит к проверке улучшений: что изменить, какой эффект ожидается, кто делает, где и когда проверить результат. Изменение режима контроля фиксируется отдельно.

Выбор показателей и порядок проверки — в плане мониторинга. Определения — в метриках, сбор — в гайде ИИ-инженера, согласование и решения — в гайде РП.

Как использовать результат

На Подготовке зафиксируйте текущие условия продукта. На PoC актуализируйте их по пробам и отдельно отметьте временные системы и отличия Пилота. Специалисты продукта получают обучение и демонстрацию; самостоятельная работа с агентами на PoC не является условием этого чек-листа. При переходе к Пилоту проверяются изменившиеся процессы и подключения.

Сначала выделяются пробелы, мешающие ближайшему рабочему пути. Для них выбирается ограничение области, временное участие человека или необходимая доработка. Остальные улучшения включаются в существующий план развития с ожидаемым результатом, ответственным и зависимыми задачами.

Незакрытый пункт не останавливает внедрение целиком. Но запуск конкретной задачи требует доступных инструментов, достаточной проверки и понятной ответственности. Например, отсутствие виртуального времени не мешает начать с задач без таймеров; отсутствие автоматических миграций допускает временный контролируемый порядок. Сократить ручную регрессию до работающей автоматической замены нельзя.

Условия команды кабинета «Меридиан Сервис»

Структура по шаблону чек-листа. Состояния пунктов, подтверждения и работы развития пока не заполнены.

Паспорт: проект кабинета. **Этап, дата оценки и версия:** … . **Область задач и системы:** … .

1. Постановка и движение задач

…

2. Требования и проверяемость результата

…

3. Сборка, данные и DEV/UAT-процесс

…

4. Управляемые зависимости и выполнение сценариев

…

5. Тесты, доказательства и диагностика

…

6. Участие людей и улучшение работы

…

7. Метрики и обратная связь

…

Результат оценки

Препятствия ближайшему пути: … . **Временные ограничения:** … . **Работы развития в трекере:** … . **Следующая оценка:** … .

Журнал улучшений агентов, процессов, инструментов и тестов

Журнал сохраняет выполненные работы по инструкциям, навыкам, контексту, процессам, инструментам, тестам и тестовым средам, а также результаты проверки изменений. Он связывает наблюдение, действие и полученный эффект. Будущие работы остаются в трекере, а рабочие значения метрик — в журнале показателей.

Форма примера для кабинета «Меридиан Сервис».

Шаблон

1. Область и источники

Продукт, паспорт и этап: … . **План мониторинга и версия:** … . **Источники работ и проверок:** … . **Период записей и часовой пояс:** … .

2. Выполненные работы и проверки

ID записи / дата	Наблюдение; задача и подтверждения	Что изменено и сделано	Ответственный; фактические исполнители и команды	Почему работу выполнил ИИ-инженер внедрения	Версия / область применения	Проверка и фактический результат	Решение и следующая работа
…	…	…	…	…	…	…	…

Указывайте фактически выполнивших изменение специалистов и их команды, даже если ответственный за работу — другой человек. Если улучшение выполнил ИИ-инженер внедрения, в соответствующей колонке поясните, почему его не выполнил ИИ-инженер продукта: например, он ещё осваивает инструмент или потребовалась помощь в сложной диагностике. Если инженер внедрения только помогал, различите исполнение и помощь. Для других случаев колонка неприменима. Совмещение назначений отражается явно.

Журнал охватывает улучшения процессов, инструментов, тестов, данных, сред, агентов, навыков и знаний. Незавершённая идея может оставаться наблюдением или работой в трекере; выполненные изменения и их проверки записываются здесь. Не нужно заранее составлять отдельный полный план всех будущих доработок.

Для проверки сохраняются условия до и после, сценарии и ревизии, ссылки на прогоны и применимые метрики из плана мониторинга. Ожидаемый эффект указывается отдельно от полученного результата. Если проверка продолжается, это отмечается со сроком и ответственным; улучшение ещё не считается подтверждённым.

Решения: сохранить / доработать / отменить изменение / вернуть прежний режим; основание и ссылка на принятое решение … . При изменении контроля указать согласованную область HITL/HOTL и условия возврата. Наличие записи не даёт разрешения ослабить проверку автоматически.

3. Исправления записей

Дата, ID, причина уточнения, прежняя и новая версия, затронутые выводы: … . Завершение проверки позже связывается с исходной работой; сама работа не считается выполненной второй раз.

Как вести

Агент собирает записи ежедневно из задач, MR, обсуждений, версий навыков и результатов проверок по порядку сбора. Новая запись нужна при выполненном действии, результате проверки или принятом решении; пустые строки за дни без изменений не создаются. Повторный сбор обновляет сведения с сохранением истории.

ИИ-инженер уточняет техническую причину, выполненное изменение и результат проверки. РП обеспечивает выполнение согласованных работ; предметные участники подтверждают проверки своей области. Один ответственный назначается на конкретную работу, даже при участии нескольких команд.

Тонкое или противоречивое наблюдение сначала может остаться фактом для накопления статистики. Оно не объявляется дефектом навыка и не превращается автоматически в обязательное исправление. Выбор группы проблем и решение об улучшении принимаются в циклах управления эффективностью. На PoC и Пилоте не требуется улучшать то, что уже обеспечивает согласованный результат, или устранять весь HITL.

Улучшения при внедрении в кабинет «Меридиан Сервис»

Структура по шаблону журнала улучшений. Записи выполненных работ и результаты проверок пока не заполнены.

1. Область и источники

Паспорт: проект кабинета. **Этап:** … . **План мониторинга:** план кабинета, версия … . **Источники, период и часовой пояс:** … .

2. Выполненные работы и проверки

ID записи / дата	Наблюдение; задача и подтверждения	Что изменено и сделано	Ответственный; фактические исполнители и команды	Почему работу выполнил ИИ-инженер внедрения	Версия / область применения	Проверка и фактический результат	Решение и следующая работа
…	…	…	…	…	…	…	…

3. Исправления записей

…

Ежедневный журнал решения задач

Журнал показывает, что происходило с продуктовыми задачами за день: работу агентов и людей, результаты, проверки, возвраты и ожидания. Его формирует агент из трекера, Git, чата, журналов агентов, CI и тестовых сред по правилам сбора. Это представление рабочих сведений, а не второй ручной трекер.

Форма примера для кабинета «Меридиан Сервис».

Шаблон

1. Период и источники

Продукт и паспорт: … . **Дата/период, этап и часовой пояс:** … .

Версии ЖЦ задач, правил Git и плана мониторинга: … . **Источники, время сбора и версия навыка:** … . **Покрытие и ограничения:** … .

2. Работа по задачам

Одна строка — задача с наблюдавшейся за день работой или открытым ожиданием. Несколько запусков, MR и исполнителей связываются с этой задачей, не превращая её в несколько результатов. Для открытого ожидания без новых событий показывается состояние на срезе; отсутствие событий в недоступном источнике не означает отсутствия работы.

Задача и версия требований	Что сделано и получено за день	Фактические исполнители: агенты и люди, роли/команды	Проверенная ревизия, среда и результаты	Ожидания, возвраты и открытые вопросы	Первичные подтверждения
…	…	…	…	…	…

Для ожидания указываются причина, начало/конец либо открытое состояние и возраст, адресат или ресурс. Различаются очередь HITL, активная проверка, зависимости, лимит параллелизма, готовность среды и инфраструктурный отказ. Параллельные интервалы не складываются автоматически в длительность задачи; правила ожиданий.

Исходы: продолжена / передана человеку / отменена / получен и принят результат — с подтверждениями по ЖЦ задач. Публикация MR или зелёный тест не заменяют приёмку; день без выпуска может содержать выполненную работу.

3. Расчётные сведения

Ссылка на набор данных для метрик и применённую методику: … . **Оценки объёма и человеческого участия, основания, статус проверки РП:** … . **Что пока неизвестно:** … .

Объём задачи не повторяется как выполненный в каждом дне. Зарегистрированные факты, оценочные часы и отсутствующие сведения различаются. Время ожидания не считается активным трудом; труд после передачи человеку сохраняется. Итоговые показатели вычисляются по плану мониторинга, их значения представлены в журнале показателей.

4. Уточнения и следующие действия

Какие сведения требуют проверки: … . **Кто и что подтвердил или исправил:** … . **Ссылки на решения и работы в трекере:** … .

Изменения агентов, навыков, процессов, инструментов и тестов отражаются в журнале улучшений. Здесь сохраняется связь с исходной задачей и наблюдением, без повторного описания улучшения.

Как вести

Сбор выполняется раз в сутки за прошлый день по расписанию либо просьбе агенту. Пропущенные дни и повторный расчёт задаются датой или периодом. Агент обновляет прежние записи с сохранением версии и причины; повторная доставка события не размножает действия, а пересборка не прибавляет тот же объём повторно. Открытые ожидания переходят между днями с исходной датой начала.

ИИ-инженер проверяет связи и извлечение событий, РП — оценочные значения и существенные расхождения, предметные специалисты — спорные результаты своей области. На PoC записывается фактическое участие ИИ-инженера; его проверки не выдаются за самостоятельную работу производственной команды. На Пилоте видны непосредственные действия специалистов и помощь внедрения.

Для воспроизведения расчёта сохраняются ссылки на первичные сведения, версии подготовленного набора и правила. Нет данных — это ограничение, а не ноль и не доказательство успеха. Команда не заполняет отдельные таймшиты для этого журнала.

Решение задач кабинета «Меридиан Сервис» за день

Структура по шаблону ежедневного журнала. Даты, выполненные действия, оценки и результаты пока не заполнены.

1. Период и источники

Паспорт: проект кабинета. **Дата/период, этап и часовой пояс:** … . **Версии процессов и плана мониторинга:** … . **Источники, время сбора, версия навыка и ограничения:** … .

2. Работа по задачам

Задача и версия требований	Что сделано и получено за день	Фактические исполнители: агенты и люди, роли/команды	Проверенная ревизия, среда и результаты	Ожидания, возвраты и открытые вопросы	Первичные подтверждения
…	…	…	…	…	…

3. Расчётные сведения

…

4. Уточнения и следующие действия

…

Журнал показателей внедрения

Журнал сохраняет результаты расчёта метрик из плана мониторинга продукта. Он показывает значения за выбранный период, полноту данных и основания выводов. Здесь нет собственного обязательного набора показателей: состав, единицы, границы и правила берутся из указанной версии плана.

Форма примера для кабинета «Меридиан Сервис».

Шаблон

1. Период и основания записи

Продукт и паспорт: … . **Этап:** PoC / Пилот. **Вид записи:** недельная сводка / итоговый срез Пилота.

Период, когорта и отсечка: … . **Часовой пояс и дата сборки:** … . Различайте период работы, окно оценки когорты и момент получения данных.

План мониторинга — ссылка и версия: … . **Набор данных и версия расчёта:** … . **Версии источников, оценивания и подтверждения РП:** … .

2. Рассчитанные показатели

ID и название метрики из плана	Область / период / когорта	Единица	База сравнения	Действующий ориентир	Рассчитанное значение	Полнота и ограничения	Подтверждение расчёта
…	…	…	…	…	…	…	…

Показываются все метрики, предусмотренные планом для этого периода. Для каждой — ссылка на её определение в плане; для разрезов создаются отдельные строки. Неизвестное значение обозначается «нет данных» с причиной, а неприменимость — с основанием. При отсутствии ориентира пишется «наблюдаем без порога». Прежние и новые редакции ориентиров не смешиваются.

Подробности полноты: количество наблюдений и покрытие, незавершённые задачи и ожидания с возрастом, пропуски, отмены и передачи людям … . Длительности ожидания, активные часы и расход платформы не смешиваются. Детальные таблицы и первичные сведения доступны по ссылкам; повторно вводить их вручную не требуется.

3. Проверка сведений и исправления

Рассмотренные аномалии и оценки: … . **Результат проверки РП и предметных участников:** … . **Что исправлено в данных или расчёте:** … . **Что осталось неподтверждённым и как это ограничивает вывод:** … .

При пересборке сохраняются предыдущая версия, причина, затронутые периоды и ссылка на новую запись. Уточнение оценки объёма задачи не объявляется приростом производительности.

4. Выводы и действия

Что показывают значения относительно цели и ограничений: … . **Какие препятствия требуют решения:** … . **Решения и работы в трекере:** ссылка, один ответственный, срок/следующая проверка … . **Следующий разбор:** … .

Результаты конкретных улучшений берутся из журнала улучшений, а не пересказываются заново. Изменение методики или ориентира оформляется в плане мониторинга с историей согласования; запись журнала ссылается на эту редакцию.

5. Итоговая запись Пилота

Для обычной недели этот раздел не заполняется. Для итогового среза указываются:

- применённая версия целевых показателей, итоговая область и окно наблюдения;
- достигнутые значения и ссылки на строки таблицы, подтверждения исходных сведений и оценок;
- состояния условий чек-листа завершения Пилота с подтверждениями;
- зафиксированный РП итог; при невыполнении условий — принятое Спонсором решение о продолжении, изменении условий с продолжением либо остановке;
- оставшиеся обязательства и ограничения, ссылки на актуальные процессы, инфраструктуру и обеспечение дальнейшей работы.

При достижении согласованных показателей и выполнении обязательных условий РП фиксирует успешное завершение. Отдельное утверждение успеха Спонсором не требуется. Обычная недельная сводка сама по себе не является итоговым решением.

Как вести

Агент формирует запись из сохранённых фактов и подтверждённых оценок по правилам автоматического сбора. РП проверяет оценки и существенные расхождения, ИИ-инженер исправляет ошибки извлечения и вычислений; роли могут совмещаться. Состав метрик не выбирается заново при каждой сборке. Если нужная метрика ещё не вычисляется, строка остаётся видимой с ограничением.

Сведения собираются ежедневно; сводка формируется еженедельно по ритму плана мониторинга. Пересборка использует те же источники, устраняет дубли и сохраняет версии. На PoC пробная запись проверяет процедуру, на Пилоте показывает рабочие результаты; эти режимы отмечаются явно. Общий разбор и выбор действий описаны в гайде управления эффективностью.

Показатели внедрения в кабинет «Меридиан Сервис»

Структура записи по шаблону журнала показателей. Значения, периоды, согласования и выводы пока не заполнены.

1. Период и основания записи

Паспорт: проект кабинета. **Этап и вид записи:** … . **Период, когорта, отсечка, часовой пояс:** … .

План мониторинга: план кабинета, применённая версия … . **Набор данных, версия расчёта и подтверждения:** … .

2. Рассчитанные показатели

ID и название метрики из плана	Область / период / когорта	Единица	База сравнения	Действующий ориентир	Рассчитанное значение	Полнота и ограничения	Подтверждение расчёта
…	…	…	…	…	…	…	…

Подробности полноты: … .

3. Проверка сведений и исправления

…

4. Выводы и действия

…

5. Итоговая запись Пилота

…