

Техническое задание. ИИ-агент для сопровождения сквозных автотестов в конвейере непрерывной интеграции

ТОО «Xena Soft»

Сентябрь 2026

1. Заказчик и контекст

ТОО «Xena Soft» (далее — заказчик) — сервисная ИТ-компания, участник Astana Hub. Разрабатывает веб-продукты по заказу клиентов из США и Европы; для каждого проекта формирует отдельную команду.

Xena Soft самостоятельно разработала и использует рабочий прототип в виде двух конвейеров непрерывной интеграции (CI) на Azure DevOps Pipelines и GitHub Actions, в которых ИИ-агент на базе Claude Code сопровождает сквозные автоматизированные тесты (далее — E2E-тесты) на Playwright. Прототип внедрён в двух проектах Xena Soft, но привязан к их инфраструктуре и соглашениям, принятым в этих проектах. Права на прототип принадлежат Xena Soft.

После выбора исполнителя и подписания соглашения о неразглашении (NDA) заказчик передаёт обезличенную копию прототипа. Клиентский код и данные заказчик передаёт только при наличии документально подтверждённого разрешения правообладателей.

2. Цель

Создать переносимый продукт «QA-агент»: единое ядро и адаптеры для трёх CI-систем. Продукт подключается к новому проекту за один рабочий день и после подключения сам сопровождает E2E-тесты: запускает их, разбирает падения, готовит исправления и расширяет покрытие. Все

изменения кода агент оформляет как запросы на слияние (PR/MR); их принимают только после ревью сотрудника заказчика.

Термины в документе: «прогон» — один запуск набора тестов на стенде; «стенд» — развёрнутое окружение продукта (dev, testing, staging, production); «боевое окружение» — production; «PR/MR» — запрос на слияние (pull request / merge request) в GitHub, GitLab или Azure DevOps; «задача» — issue или work item в трекере.

3. Что должен делать агент

3.1. Запуск

- Триггеры: завершение развёртывания (веб-хук или событие CI), расписание, ручной запуск с параметрами «окружение» и «репозиторий».
- Для каждой пары «репозиторий + окружение» одновременно идёт не больше одного прогона, а в очереди ждёт только последний. Текущий прогон можно отменить, пока агент не начал вносить изменения во внешние системы. После создания ветки, задачи или тестовых данных агент завершает начатые операции или отменяет внесённые изменения.
- Перед прогоном агент проверяет, доступна ли LLM. Если квота исчерпана, а время до её сброса превышает порог ожидания, прогон завершается со статусом `usage limit` (нет доступа к LLM), а не тратит час на тесты впустую.

3.2. Обработка результатов прогона

Агент классифицирует результат и выбирает действие. Если причину определить нельзя, а также если упал стенд или сеть либо не хватает прав или данных, агент завершает прогон со статусом `blocked`, описывает, что нужно уточнить, и не меняет код.

Результат	Режим	Что делает агент
Тесты упали	Разбор падения и исправление	Читает лог, HTML-отчёт Playwright, трейсы, консоль и сетевые запросы. Отдельным шагом определяет источник сбоя — тест, продукт или инфраструктуру — и соответствующий репозиторий.

Результат	Режим	Что делает агент
		Устаревший тест исправляет и открывает PR/MR (статус <code>fix</code>). Предполагаемый дефект продукта оформляет как задачу, приводит подтверждающие материалы и указывает вероятное место дефекта в коде. Нестабильный тест помечает и отдельно указывает в отчёте.
Тесты прошли	Расширение покрытия	Изучает изменения продукта, внесённые после последнего обработанного коммита или идентификатора развёртывания, и связанные с ними задачи. Если находит подтверждённый недостаток покрытия, пишет сценарий, прогоняет его на стенде до коммита и открывает PR/MR (статус <code>coverage</code>). Проверяет открытые PR/MR, чтобы не создавать дублирующие сценарии. Если недостатков нет, фиксирует результат <code>green</code> (изменения не требуются).
Тестов нет	Создание набора с нуля	Создаёт набор по утверждённому для проекта перечню: авторизация для заданных ролей, ключевые пользовательские сценарии, переходы между разделами или приложениями продукта (статус <code>bootstrap</code>). Если элементы интерфейса нельзя надёжно идентифицировать, готовит отдельный PR/MR со стабильными селекторами и останавливается (статус <code>selectors</code>); тесты создаёт во время следующего прогона, после развёртывания. Семантические атрибуты (<code>role</code> , <code>aria-label</code>) добавляет только там, где они верно описывают назначение элемента; технические селекторы оформляет как <code>data-testid</code> по соглашениям проекта.

3.3. Источник ожидаемого поведения

Для каждого проекта заказчик указывает в конфигурации источники ожидаемого поведения: требования или критерии приёма, задачи трекера, спецификации API, существующие тест-планы и перечень критических сценариев. Если источников нет или они противоречат друг другу, агент не меняет проверки и завершает прогон со статусом `blocked`.

3.4. Правила, которые нельзя нарушать

- Агент не ослабляет проверки ради успешного прогона и не изменяет продукт так, чтобы он соответствовал неверному тесту.
- Агент не сливает ветки и ничего не развёртывает. Работа с боевыми окружениями по умолчанию запрещена. Её можно включить отдельно, только для утверждённого перечня тестов, которые не меняют состояние продукта, с минимально необходимыми правами и без создания данных.
- Тест, воспроизводящий текущее поведение продукта, проходит даже при наличии дефекта и тем самым может закрепить дефект как норму. Если поведение приложения внутренне противоречиво (валидация не срабатывает, лимит объявлен в интерфейсе, но не применяется), агент не добавляет тест для такого поведения и описывает предполагаемый дефект в PR/MR.
- В каждом PR/MR агент указывает, что было фактически проверено, а не только что требовалось проверить, и ставит статус проверки: `pass` / `fail` / `blocked` / `none`.

3.5. Работа с кодом

- У агента одна долгоживущая ветка на репозиторий и один накопительный PR/MR, который пополняется с каждым прогоном: иначе каждый прогон не видел бы ещё не принятые изменения предыдущего и делал бы ту же работу заново. Операции записи упорядочиваются на уровне репозитория, а не пары «репозиторий + окружение».
- Переписывать историю ветки запрещено. Если ветка агента разошлась с базовой, агент заново применяет свои изменения поверх последнего коммита базовой ветки и повторно запускает затронутые тесты.
- Перед открытием PR/MR агент запускает автоматические проверки только для внесённых им изменений.

- Все операции записи в GitHub, GitLab и Azure DevOps выполняются через выделенное приложение или бота с короткоживущим токеном, а не через токен конвейера.

3.6. Тестовые данные

- Всё, что создаёт прогон, называется по шаблону `autoqa-<run_id>-<slug>`; идентификаторы созданных объектов агент записывает в реестр.
- На этапе очистки набор тестов удаляет созданные во время прогона данные через API продукта, без браузера.
- Ночная очистка удаляет только объекты из реестра, только в явно разрешённых небоевых окружениях, только старше заданного срока и не больше заданного числа за запуск. Перед удалением агент делает запись в журнале. Если у продукта нет подтверждённого API очистки, режим создания данных для него отключён.

3.7. Доступ к LLM

- Для доступа используются организационные или сервисные учётные данные, разрешённые провайдером. Агент умеет переключаться между согласованными провайдерами и проектными квотами, но не обходит ограничения провайдера.
- Задаются лимиты на стоимость, число запросов и время выполнения одного прогона.
- Учётные данные обновляются средствами провайдера и хранилища секретов CI: все операции журналируются, а одновременное обновление блокируется; значения в лог не выводятся.
- Запрет рекурсии: один запуск агента не может инициировать другой.
- Проверочный запуск без обращений к LLM — для отладки маршрутизации конвейера.
- Реализация не зависит от конкретного провайдера: Claude Code можно заменить другим агентным CLI без правки ядра.

3.8. Безопасность

- Код репозитория, содержимое веб-страниц, задачи, комментарии, названия тестов и вывод модели — недоверенные данные: они не могут менять системные правила агента. Агент работает в изолированном окружении с минимальными правами и списком

разрешённых команд и сетевых адресов; все внешние операции регистрируются в журнале аудита.

- Всё, что агент выводит в CI, экранируется по правилам конкретной CI-системы, чтобы текст из внешнего источника не был воспринят как управляющая команда.
- Секреты не попадают в переменные окружения конвейера, в артефакты и в задачи. На этапах, требующих ввода пароля оператора, запись трейсов и видео отключается.
- В задачах нет ссылок на трейсы, HAR и видео.
- В конфигурации проекта записано, какие данные (исходный код, артефакты прогона, персональные данные) можно передавать LLM-провайдеру, в каком регионе он их обрабатывает и сколько хранит.

4. Адаптеры CI

Ядро — скрипты и промпты, не зависящие от платформы. Адаптеры для GitHub Actions, GitLab CI и Azure DevOps Pipelines отвечают за триггеры, блокировки, чтение и запись секретов, публикацию отчётов, создание PR/MR и задач.

Промпты хранятся отдельно от исполняемого кода, версионируются и собираются из шаблонов с проверкой параметров. Если обязательный параметр шаблона не заполнен, запуск останавливается с ошибкой, а не выполняется с некорректно сформированной инструкцией.

5. Панель и отчётность

- Веб-панель: история прогонов по проектам и окружениям, результат каждого прогона, ссылки на PR/MR, задачи и HTML-отчёт. Доступ разграничен по проектам; срок хранения истории задаётся в конфигурации. Результаты прогона: `coverage` (расширено покрытие), `bootstrap` (создан набор тестов), `fix` (исправлены тесты), `selectors` (добавлены селекторы в продукт), `green` (изменения не требуются), `no suite` (набора тестов нет, и создать его в этом прогоне нельзя), `superseded` (прогон отменён более новым), `blocked` (прогон заблокирован), `usage limit` (нет доступа к LLM).
- Метрики: доля инцидентов, пригодных для автоматической обработки, по которым агент подготовил подтверждённый результат без ручной правки кода (знаменатель и исключения показаны рядом

с метрикой); среднее время от падения до PR/MR; число новых сценариев и покрытых ими требований в неделю; число предполагаемых дефектов.

- Для каждого прогона агент готовит понятную менеджерам продукта сводку без жаргона Playwright: что изменено, что исправлено, что предположительно является дефектом.
- Уведомления в MS Teams и Slack. Набор событий настраивается отдельно для каждого проекта. Обязательные события: PR/MR, требующий ревью; предполагаемый дефект продукта; результат blocked; исчерпание квоты LLM.
- Дополнительно: синхронизация с Azure Test Plans (создание test case, привязка по идентификатору в заголовке теста).

6. Технологические ограничения

- Тесты: Playwright, TypeScript, как минимум Chromium.
- Оркестрация: bash или TypeScript, тесты оркестрации обязательны (bats или аналог), линт shellcheck.
- Поддержка двух структур репозитория: отдельный пакет e2e/ и пакет e2e/ в составе workspace монорепо.
- Адреса стендов, права записи по окружениям, роли и тестовые учётные записи, разрешённые репозитории и API очистки описаны в одном файле конфигурации проекта, а не разбросаны по коду.

7. Результаты работы

1. Исходный код, версионизируемые установочные артефакты ядра и трёх адаптеров, схема конфигурации и пример развёртывания.
2. Панель: исходный код, схема хранения данных, аутентификация и разграничение доступа, миграции, резервное копирование, инструкция по эксплуатации.
3. Документация: подключение нового проекта с перечнем предварительных условий, описание всех режимов, инварианты и правила безопасности, перечень зависимостей и их лицензий.
4. Пилот на одном проекте заказчика: 4–6 недель, отчёт по метрикам из раздела 5.
5. Передача и обучение команды заказчика: акт и запись демонстрации.

Исходный код исполнитель передаёт в репозиторий заказчика. Стороны фиксируют в договоре объём передаваемых прав, право изменять и распространять код и допустимые лицензии сторонних компонентов.

8. Критерии приёмки

- Сотрудник заказчика по документации, за один рабочий день и без участия исполнителя подключает новый проект, который соответствует предварительным требованиям, указанным в документации.
- Контрольный набор: не менее 20 воспроизводимых случаев изменения интерфейса. Не менее чем в 80% случаев агент готовит PR/MR, в котором изменённый тест проходит и отдельно, и в составе целевого набора. Причину каждого случая и правильность результата подтверждает назначенный ревьюер. Сбои инфраструктуры, нестабильные тесты и случаи без утверждённого ожидаемого поведения в знаменатель не входят.
- На пилоте после успешных прогонов агент создаёт PR/MR только при подтверждённом недостатке покрытия, и каждый такой сценарий проходит на стенде.
- Для каждого из трёх CI-адаптеров на тестовом репозитории проверяются: ручной запуск, запуск после развёртывания, расписание, отмена устаревшего запуска, блокировка одновременной записи, получение секретов, публикация отчёта, создание PR/MR и задачи. Исполнитель фиксирует результаты в протоколе интеграционных испытаний.
- За время пилота ревью PR/MR не выявило ни одного случая, когда агент ослабил проверки или изменил продукт так, чтобы он соответствовал неверному тесту.
- За время пилота ни один секрет не попал в логи, артефакты и задачи.
- Панель показывает согласованный набор статусов, разграничивает доступ по проектам и хранит историю в течение заданного срока.
- Все скрипты оркестрации покрыты тестами, проверка линтером проходит успешно.

9. Этапы и формат взаимодействия

Этап	Содержание	Срок
1	Аудит прототипа, требования, модель угроз, архитектура ядра и адаптеров	2 недели
2	Ядро и адаптер GitHub Actions, стендовые проверки	6 недель
3	Адаптеры GitLab CI и Azure DevOps, панель, метрики	6 недель
4	Пилот на проекте заказчика	4–6 недель
5	Итоговая документация, передача, обучение команды заказчика	2 недели

Итого 20–22 недели. По согласованию этапы 3 и 4 могут выполняться параллельно; тогда общий срок сокращается до 16 недель.

Отбор исполнителя. Кандидат присылает краткое описание архитектурного подхода, предварительную оценку сроков и стоимости, состав команды и обезличенный пример похожей работы. После первичного отбора стороны подписывают NDA, и выбранный исполнитель получает обезличенную копию прототипа.

Формат взаимодействия: еженедельная онлайн-встреча, задачи в трекере заказчика, код в репозитории заказчика с ревью. Каждый этап завершается демонстрацией и письменной приёмкой. Стороны фиксируют в договоре стоимость, валюту, налоги, состав результатов каждого этапа, срок проверки, основания для отказа в приёмке и срок оплаты.