

Чек-лист пилота Codex SDK в инженерной команде

Короткая памятка для DevEx, platform engineering и security-команд: как проверить агентный coding workflow без лишнего риска для репозитория и инфраструктуры.

1. Статус и границы SDK

- Проверить актуальный README Python SDK в репозитории openai/codex.
- Не называть SDK стабильным GA-релизом без отдельного подтверждения OpenAI.
- Зафиксировать версию зависимости и план обновлений.
- Описать fallback, если API или поведение SDK изменятся.
- Разделить в документации Codex product, Python SDK, CLI-обёртки и экспериментальные примеры Agents SDK.

2. Права доступа и sandbox

- Начать пилот с read-only анализа репозитория.
- Разрешать workspace_write только после подтверждения инженера.
- Не использовать full_access в средах с секретами без отдельной оценки риска.
- Определить, какие команды агенту запрещено запускать всегда.
- Ограничить доступ к credentials, CI secrets и deploy pipelines.

3. Workflow и контроль

- Создавать отдельный thread на задачу или тикет.
- Логировать turns, streaming events и итоговые изменения.
- Запускать тесты и проверки до передачи результата в review.
- Требовать human-in-the-loop перед записью в критичные файлы.
- Хранить diff и план отката для каждого изменения.

4. Роли и ответственность

- Назначить владельца пилота со стороны DevEx или platform engineering.
- Согласовать правила с security team до подключения к реальным репозиториям.
- Определить, кто принимает pull request, подготовленный агентом.
- Обучить разработчиков читать diff и проверять архитектурные последствия.
- Не переносить ответственность за результат на модель или SDK.