ГОД ПРАКТИКИ

Агентная разработка ПО

по состоянию на ноябрь 2026 года

Данил Письменный

IT-Link · Чебоксары · 7 ноября 2026

v2.0

Данил Письменный

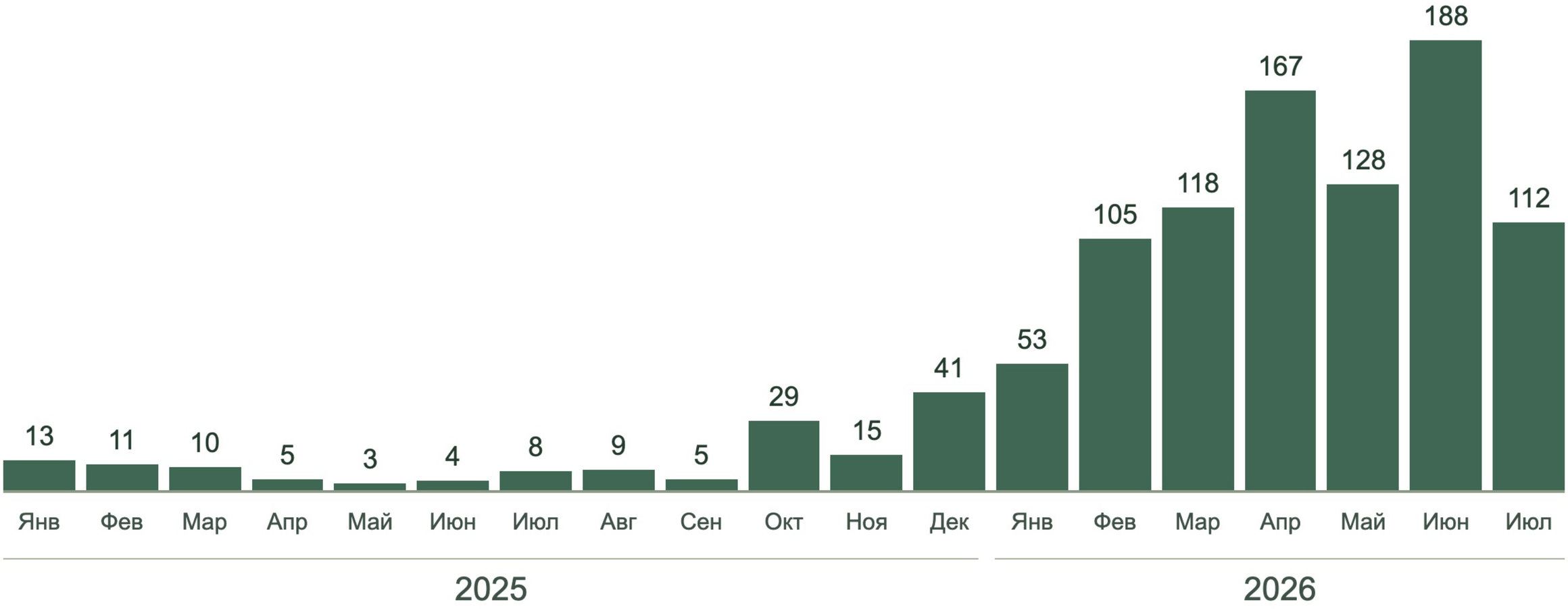
Исследую. Экспериментирую.

Строю агентные системы.



Как выглядит внедрение агентной разработки

Открытые pull requests по месяцам



21:08
ср, 4 ноября



Что я миру говорю — то и получаю

Песни для Счастья

4:21 0:00

Линьер.

Не понимаю, что ты имеешь в виду под «Линьер». Попробуй ещё раз.

Повтори что сказал но очень громко

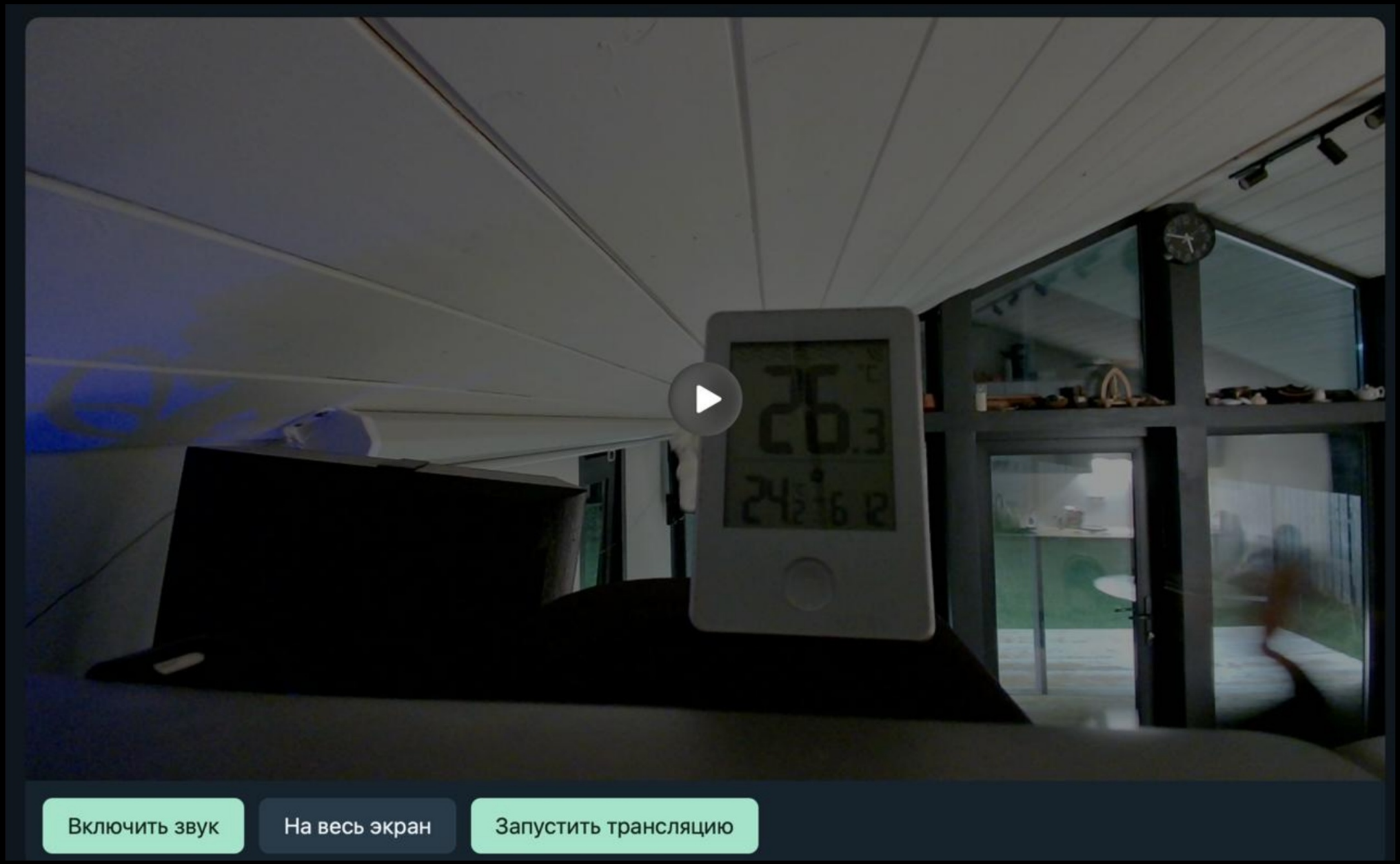
Джарвис не говорил ничего нового. Последний раз я подтверждал действие — какое именно, вспомнить не могу, это было давно.

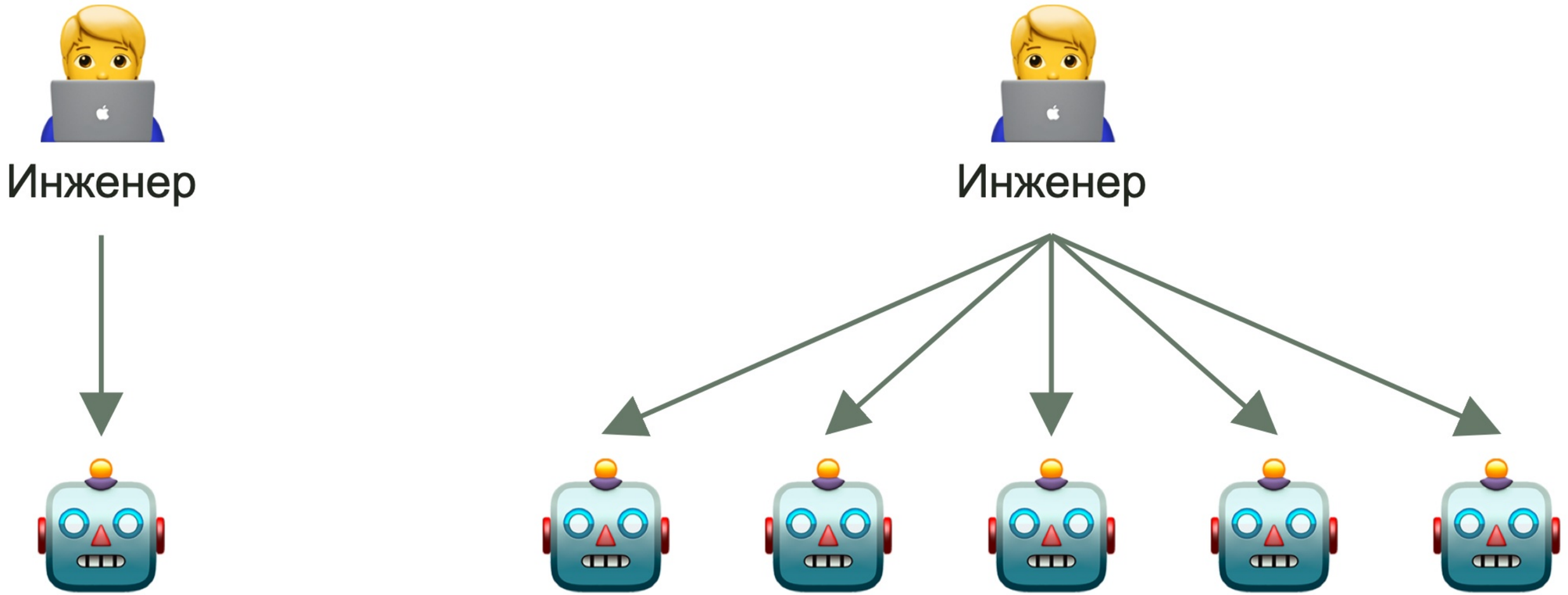
На iPad доступны журналы результатов и три совета

Результаты и советы можно открыть в полноэкранной истории; установлена и проверена версия с тремя независимыми советами.

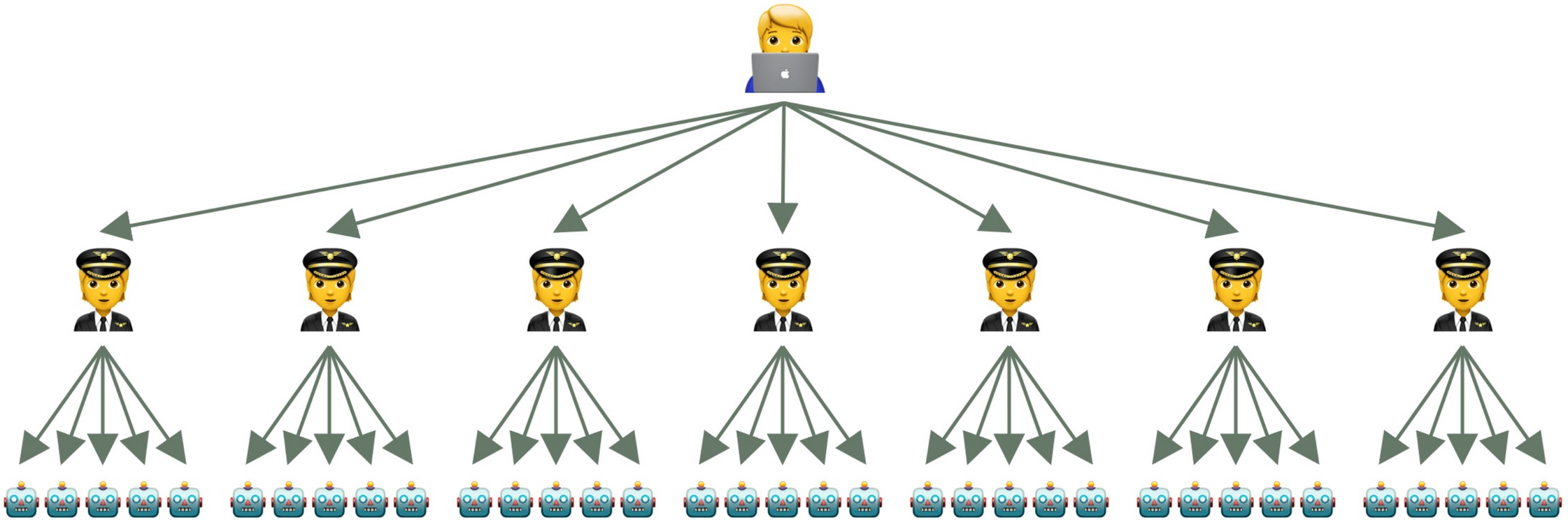
Например: включи музыку

Свет · Вода · 29° · 0° · ? · −1° · Фасад · Стоянка · Дорога · Кран · Ворота

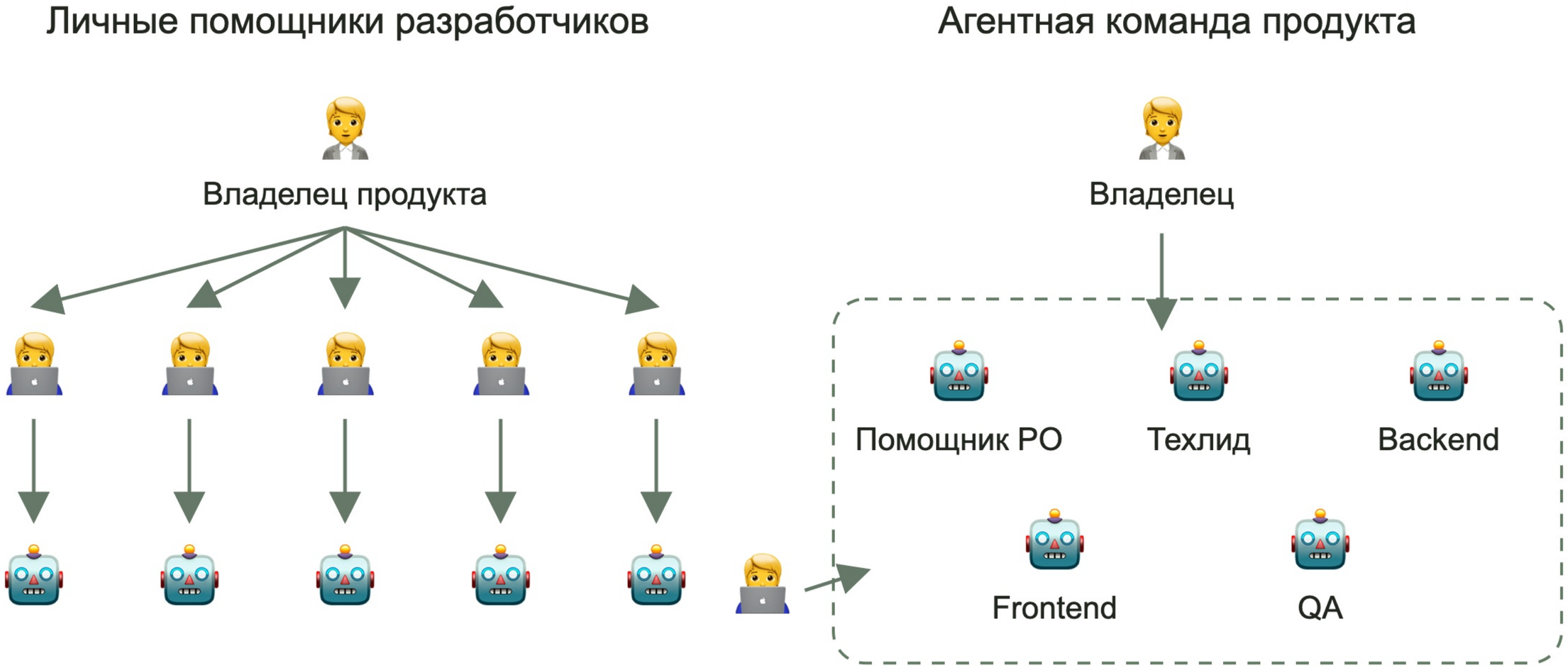

Включить звук
На весь экран
Запустить трансляцию


Инженер
Инженер

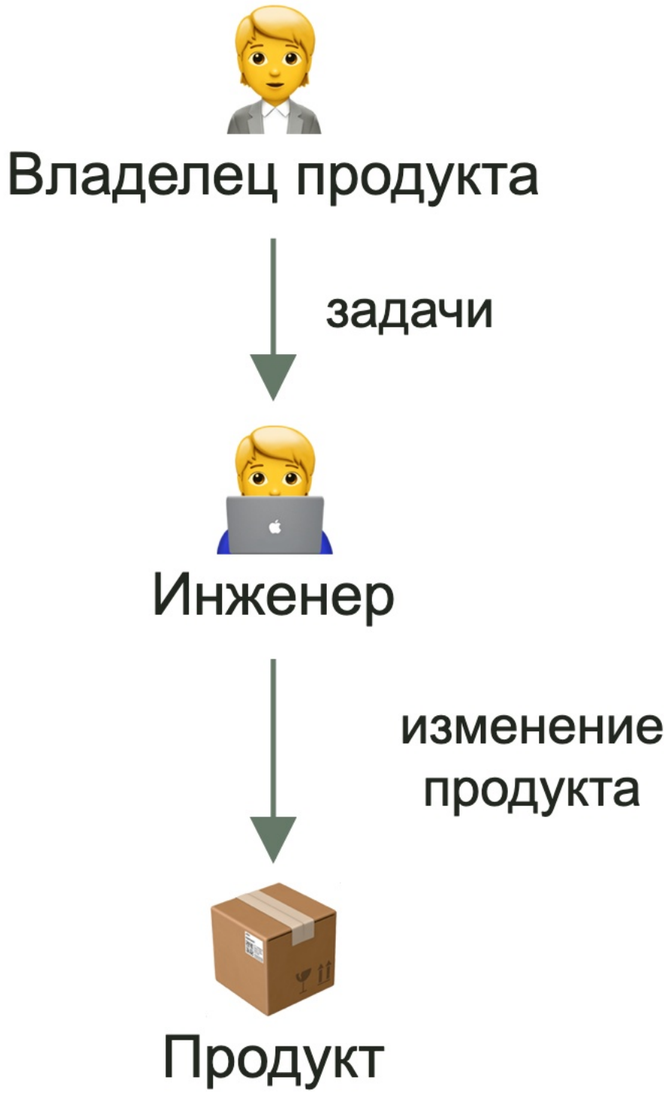
А если собрать организацию?



Два способа организовать разработку



Прямой доступ владельца продукта к изменениям



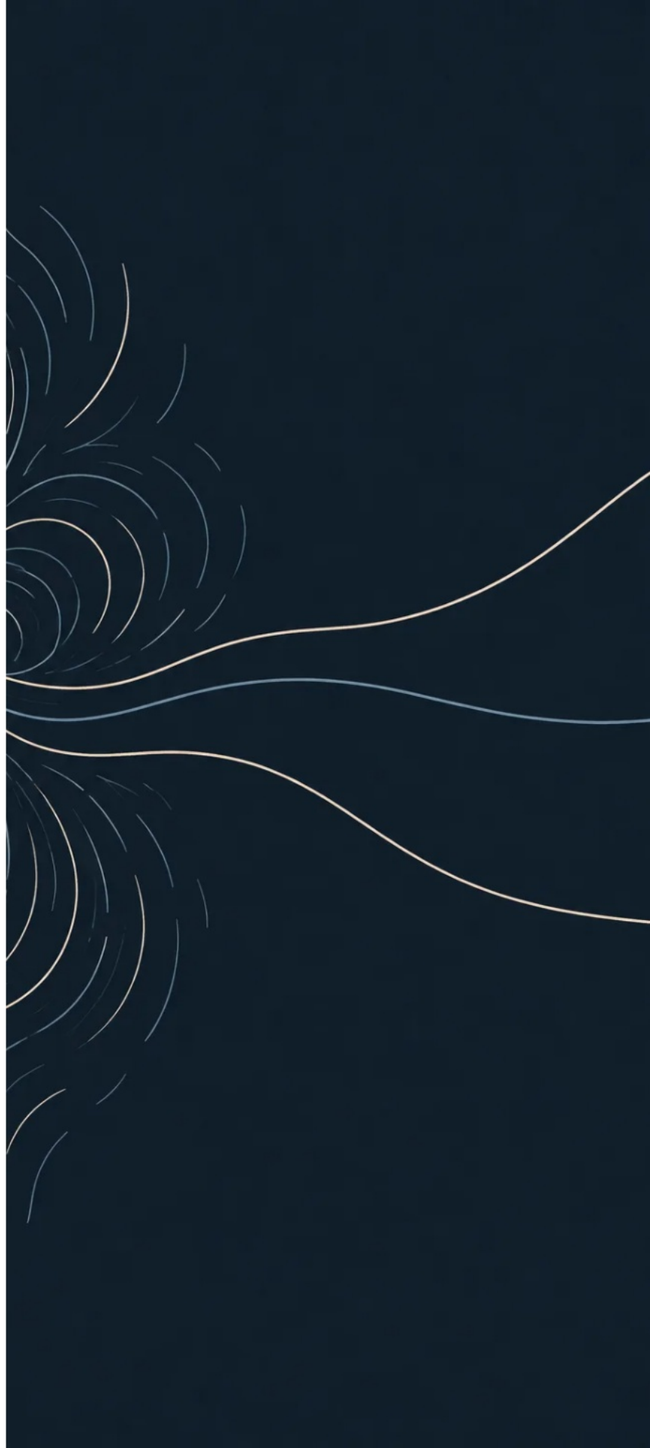


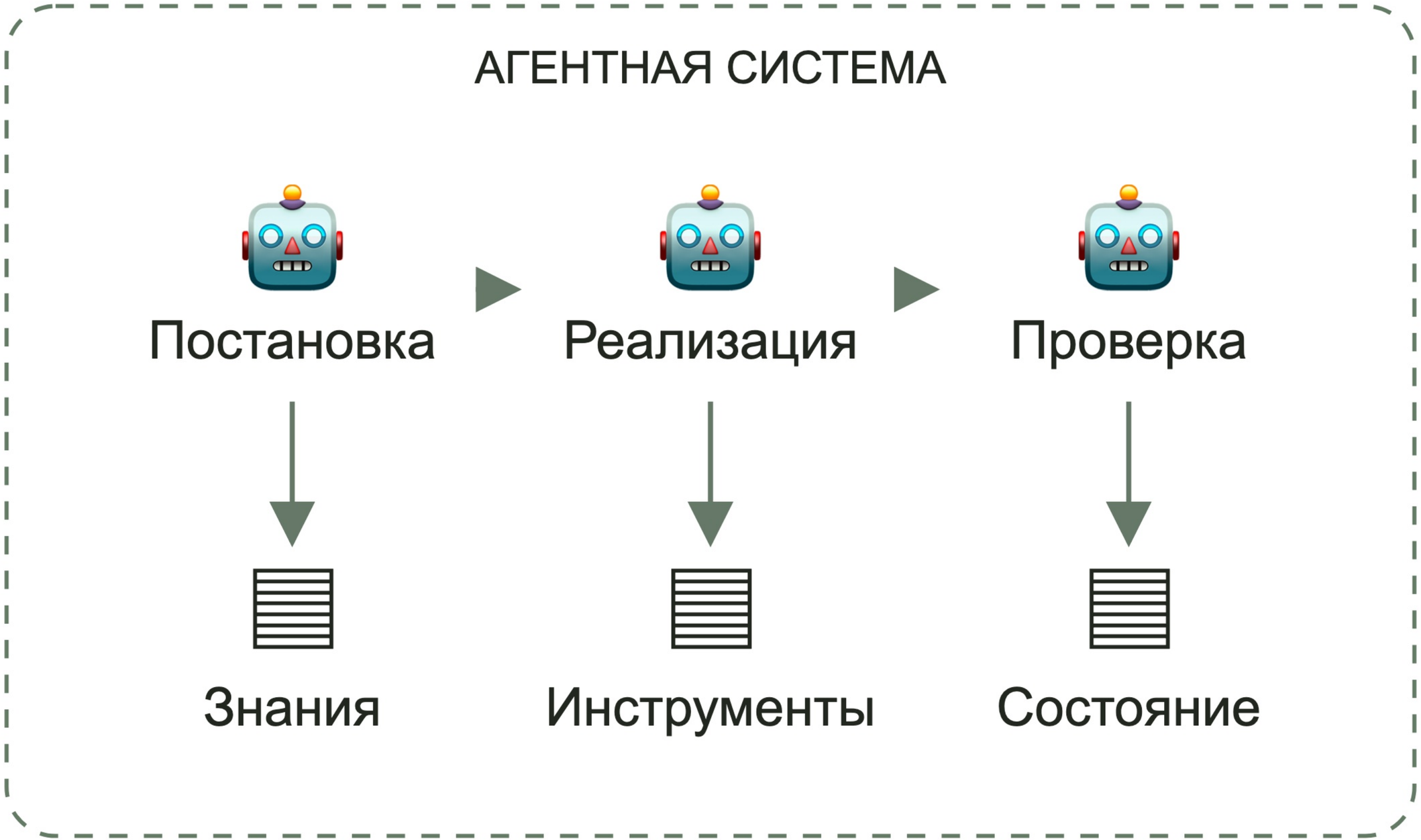
До агентной разработки — инцидент-менеджмент

Каждый инцидент:

- Зафиксирован и расследован — PIR
- Задачи предотвращения выполнены и проверены

Починили ≠ закончили




АГЕНТНАЯ СИСТЕМА
Постановка
Реализация
Проверка
Знания
Инструменты
Состояние

База знаний · 1/3

Продукт

Цели продукта
Требования
Пользователи и роли
Сценарии использования
План развития

Устройство

Архитектура
Модель данных
API и контракты
Интеграции

Работа

Инструкции
Навыки / skills
Рабочие процессы
Эксплуатационные инструкции

Качество

Критерии приёмки
Тесты и проверки
Ограничения
Безопасность и доступы

Эксплуатация

Окружения и конфигурация
Развёртывание и релизы
Мониторинг и метрики
Инциденты и разборы

Опыт

Решения
Словарь предметной области
Технический долг
Обратная связь
Память

База знаний · 2/3

Что нужно знать?

Цели продукта

Требования

Пользователи и роли

Сценарии использования

Словарь предметной области

Архитектура

Модель данных

API и контракты

Ограничения

Как действовать?

Инструкции

Навыки / skills

Рабочие процессы

Интеграции

Окружения и конфигурация

Развёртывание и релизы

Безопасность и доступы

Эксплуатационные инструкции

Как проверять и учиться?

Критерии приёмки

Тесты и проверки

Мониторинг и метрики

Инциденты и разборы

Обратная связь

Решения

План развития

Технический долг

Память

База знаний · 3/3

Понять задачу

Цели продукта

Требования

Пользователи и роли

Сценарии использования

Словарь предметной области

Ограничения

Подготовить решение

Архитектура

Модель данных

API и контракты

Интеграции

Решения

План развития

Выполнить и проверить

Инструкции

Навыки / skills

Рабочие процессы

Критерии приёмки

Тесты и проверки

Безопасность и доступы

Технический долг

Запустить и улучшать

Окружения и конфигурация

Развёртывание и релизы

Эксплуатационные инструкции

Мониторинг и метрики

Инциденты и разборы

Обратная связь

Память

База знаний: постоянные вопросы

01 Что нужно знать агентам?

02 Как организовать знания?

03 Что хранить здесь, а что — снаружи?

04 Каким источникам доверять?

05 Как искать знания и выявлять пробелы?

06 Кому какие знания доступны?

07 Кто и когда обновляет базу?

08 Как проверять новые знания?

09 Как устранять устаревшее и противоречия?

10 Как измерять пользу базы?

…и ещё десятки вопросов

Память — накопленный опыт

Опыт работы

Что пробовали?
Что сработало?
Где ошиблись?

→

Короткий путь

Как достичь цели
быстрее в
следующий раз?

→

Новая задача

Найти нужный опыт.
Добавить в контекст.
Применить.

Память — процесс

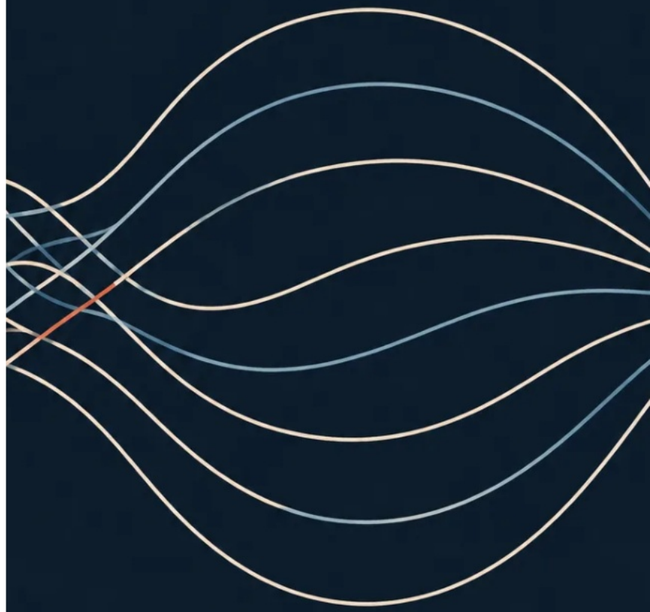
Извлекать опыт · Использовать · Пересматривать

Опыт устаревает — память нужно обновлять и чистить.

Полный цикл разработки

Доводить задачу до результата

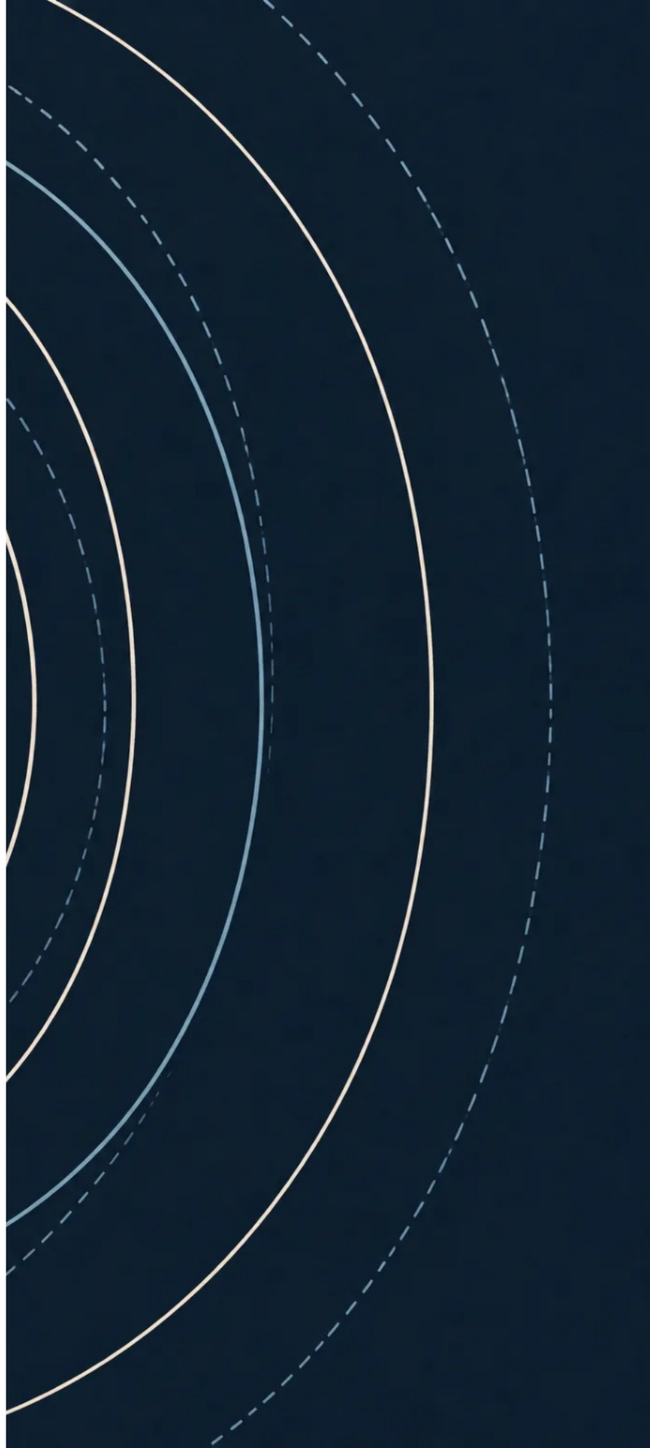
- Постановка → спецификация → план
- Реализация → проверка → выпуск
- У каждого перехода — критерий готовности



Контекст-инжиниринг

Дать агенту всё нужное для задачи

- Цель, ограничения и примеры
- Актуальные источники вместо всего архива
- Детали — по мере необходимости



Разработка через спецификации

Сначала договориться о результате

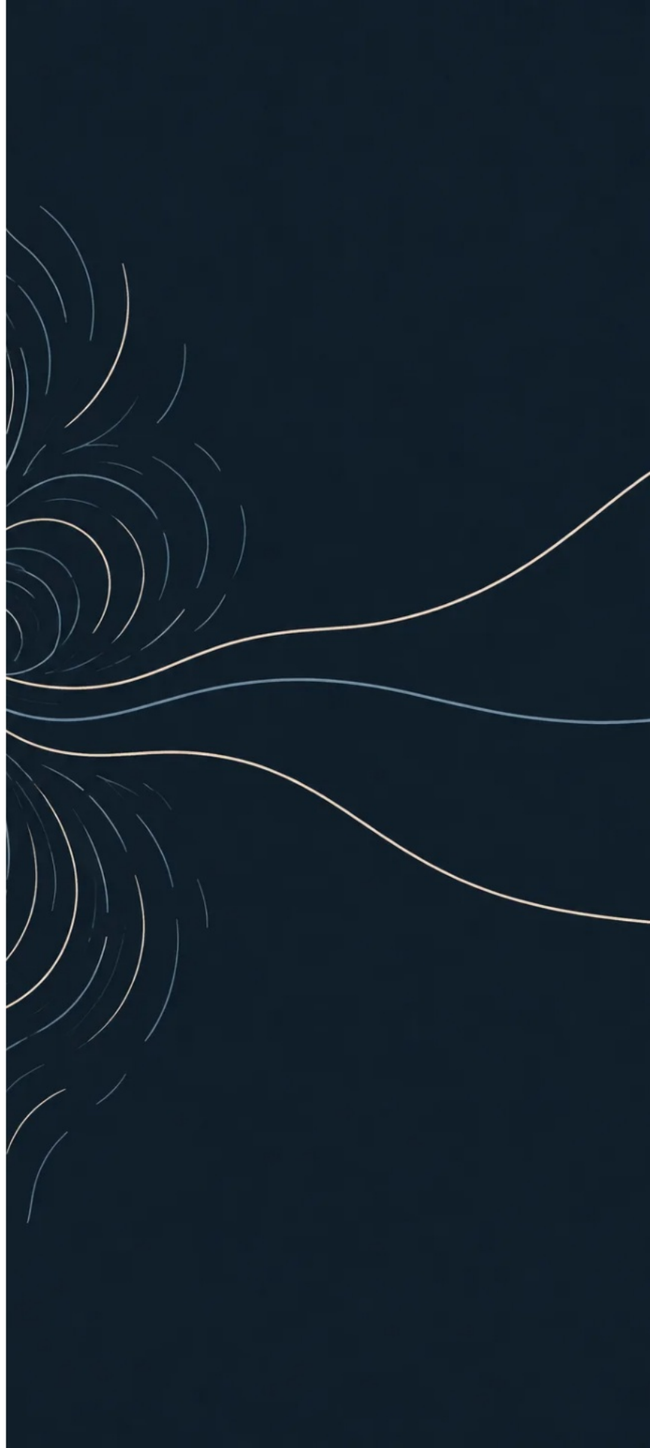
- Что должно измениться
- Что входит в задачу и что за её границами
- Как система ведёт себя в особых случаях



Критерии готовности

Заранее решить, что считать успехом

- Наблюдаемый результат
- Конкретный способ проверки
- Явные условия завершения работы



Вертикальные срезы

Одна задача — одна работающая возможность

- От действия пользователя до данных
- Небольшая область изменений
- Сквозная проверка результата



Память решений · ADR

Сохранить, почему сделали именно так

- Контекст и рассмотренные варианты
- Причины выбора и ограничения
- Связь решения с кодом и проверками



Переиспользуемые навыки

Удачный способ работы превратить в skill

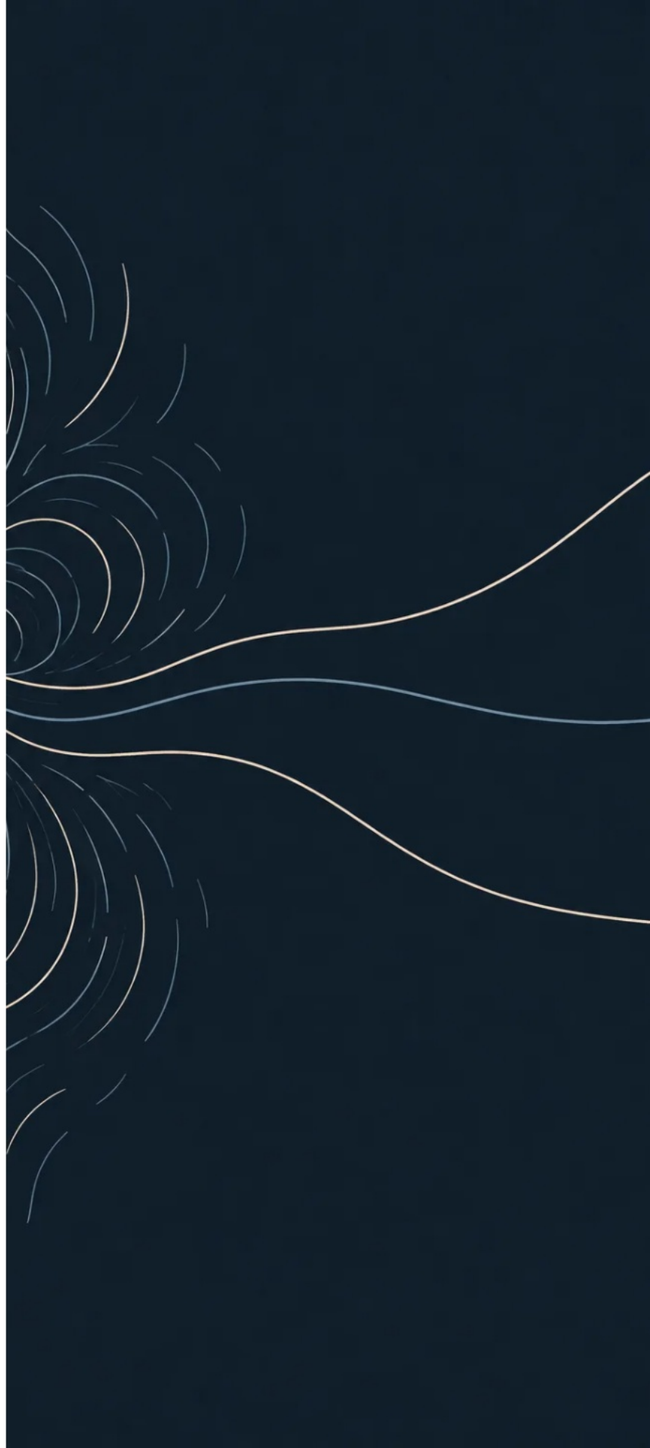
- Когда применять
- Какие шаги выполнить
- Как проверить результат



Оценка результата · eval

Принимать результат по проверочным фактам

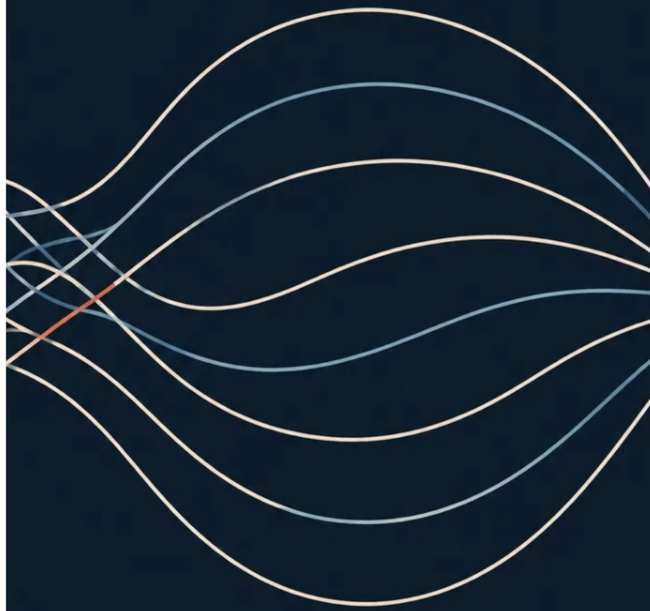
- Спецификация, код и действия агента
- Тесты, diff и наблюдаемый результат
- Принять · доработать · передать человеку



Проверка и улучшение

Замкнуть обратную связь

- Исполнить → проверить → исправить
- Давать конкретные замечания
- Ограничивать повторы без прогресса



Отладка агентного процесса

Устранить причину повторяющегося сбоя

- Восстановить, что сделал агент
- Проверить контекст, инструкции и инструменты
- Повторным прогоном подтвердить исправление



Оркестрация агентов

Разделить работу и собрать общий результат

- Исполнитель и модель под тип задачи
- Изоляция независимых задач
- Проверка и объединение результатов



Управляемая автономия

Определить границы самостоятельности

- Что агент решает сам
- Где нужна проверка человека
- Когда остановиться и передать решение



Безопасность агентов

Ограничить последствия ошибки

- Минимальные права под конкретную задачу
- Секреты — в хранилище, не в коде и чатах
- Опасные действия — с подтверждением



Среда для агентов

Проектировать условия надёжной работы

- Доступные инструменты и воспроизводимый запуск
- Автоматические проверки и ограничения
- Регулярное устранение рассогласований



Как увидеть эффект для бизнеса

АКТИВНОСТЬ

Больше агентов
Больше кода
Больше pull requests

РЕЗУЛЬТАТ

Решена задача клиента
Принята поставка
Получена оплата

За кем следить



Denis Sexy IT

Денис Ширяев · технологии и ИИ

t.me/denissexy

CodeAlive

STREAMLINE DEVELOPMENT
WITH CODEALIVE

CodeAlive-AI

FPF: системное мышление
и принятие решений с ИИ

github.com/CodeAlive-AI

уже

Всё ~~будет~~ хорошо.

СПАСИБО

pismenny.ru/it-link