Архитектура контекста и оркестрация агентов

Строим детерминированный AI-процесс

Дмитрий Путьков
Руководитель iOS команды Инвестпродукты

Дмитрий Путьков

Руководитель iOS команды
Инвестпродукты

Архитектура и оркестрация агентов



Что будет сегодня

01
Мотивация и идея

02
Архитектура системы
(KB + агенты + оркестратор)

03
Ключевые приёмы

04
Метрики

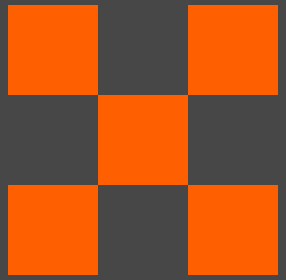
05
Где сломалось
и чему научились

Мотивация

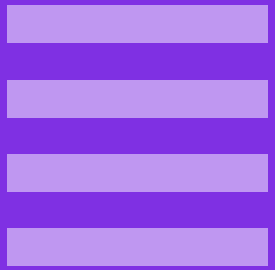


Скорость фичи ≠ скорость кода

Разработка за день — релиз через неделю



Цикл:
разработка → регресс → правки → регресс



Бутылочное горлышко не кодинг

А качество первой итерации

Шаг 1

Багфикс как проверка идеи

Почему багфикс?

01

Реальный продукт

Настоящие баги
из YouTrack

02

Чистит беклог

Конкретная единица
перфоманса

03

Тест идеи

Понимание
контекста

Workflow bug-fixer

Шаг 0

Прочитай КВ

Пойми структуру модуля

Шаг 1

Проанализируй баг

Что сломалось?

Шаг 2

Найди код

Grep по описанию

Шаг 3

Сопоставь с докой

Как должно работать?

Шаг 4

Исправь

С учётом контекста

Шаг 5

Affected_files

Что тестировать дальше

Первый буст

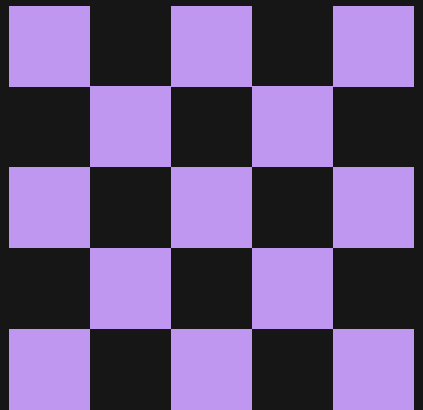
7

Багфиксов,
исправленных автоматически

30 мин

Худшая

А что если AI может забрать на себя целый процесс — не звено?



Ручной процесс

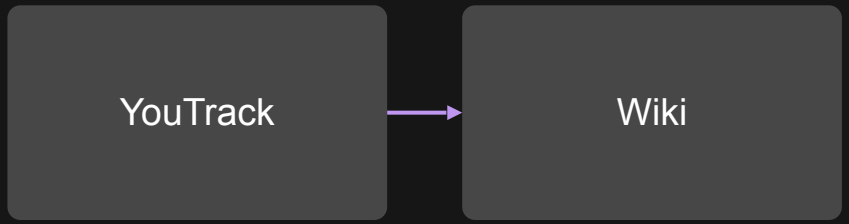
8 шагов · 8 контекстных переключений

Ручной процесс

YouTrack

8 шагов · 8 контекстных переключений

Ручной процесс



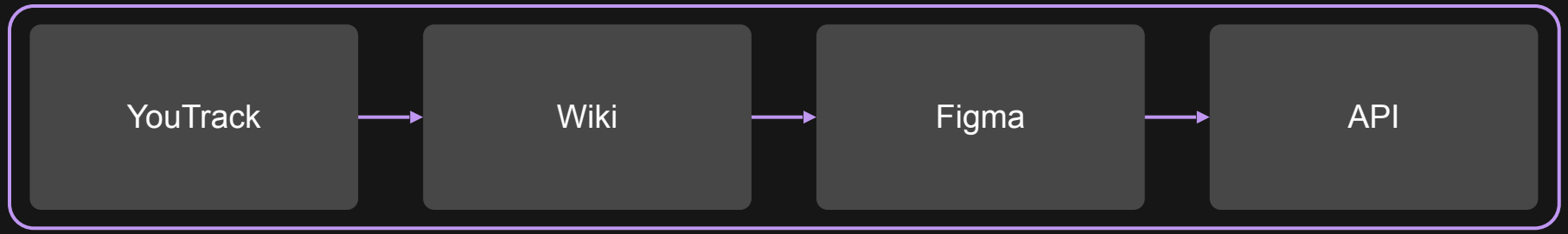
8 шагов · 8 контекстных переключений

Ручной процесс



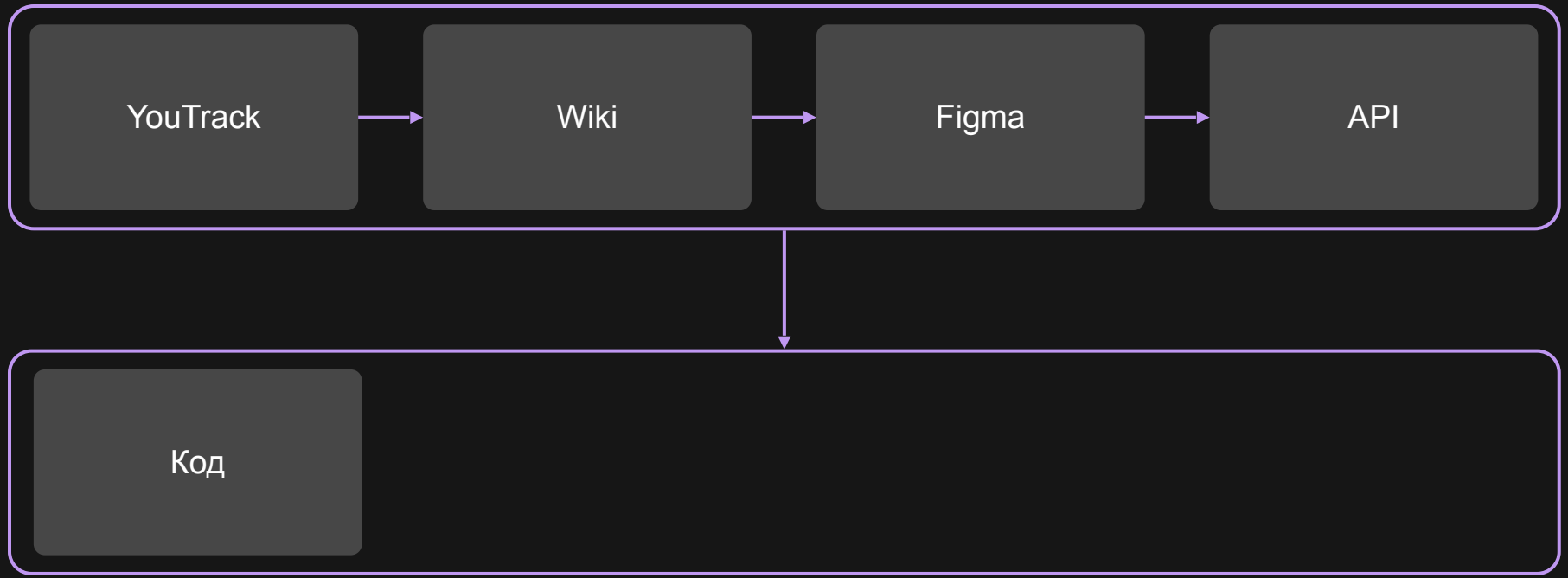
8 шагов · 8 контекстных переключений

Ручной процесс



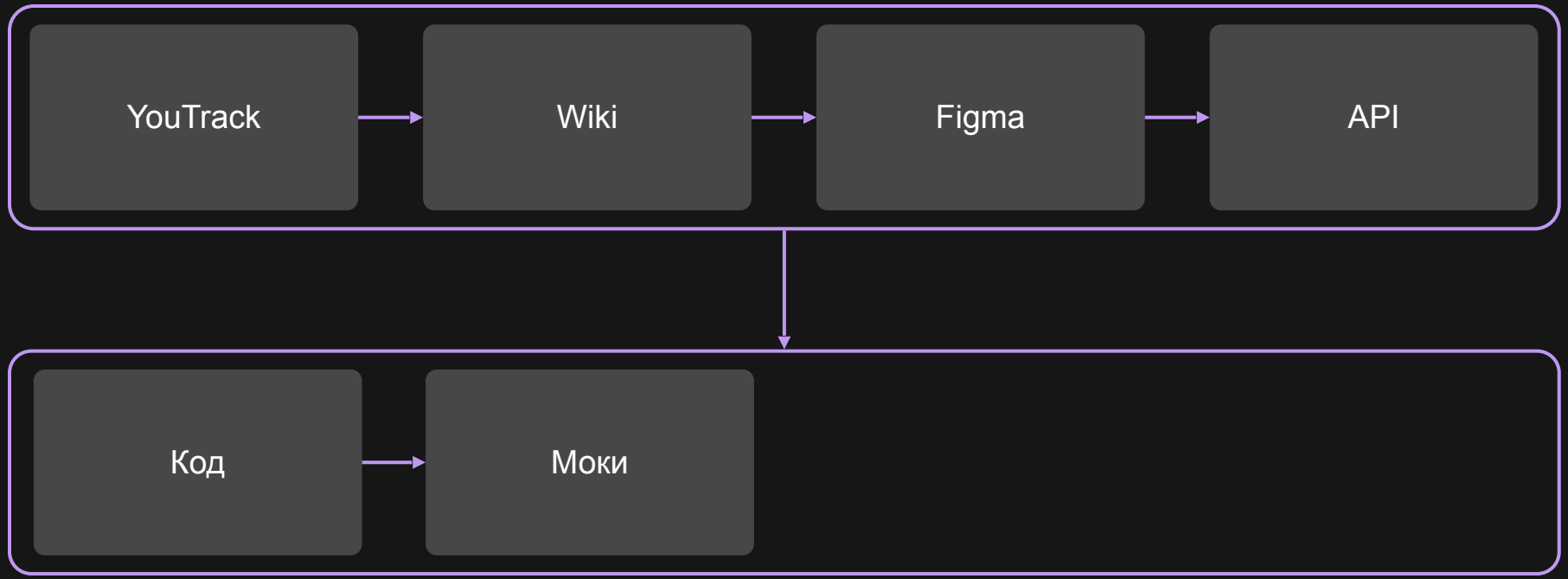
8 шагов · 8 контекстных переключений

Ручной процесс



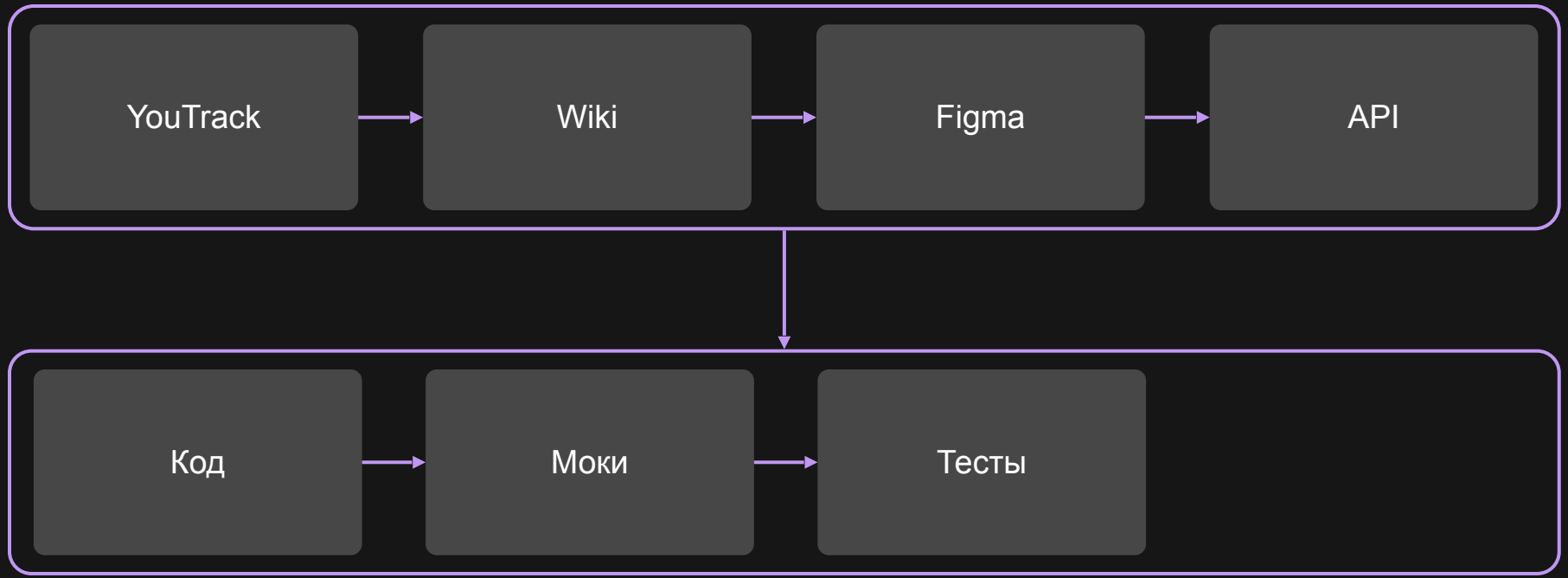
8 шагов · 8 контекстных переключений

Ручной процесс



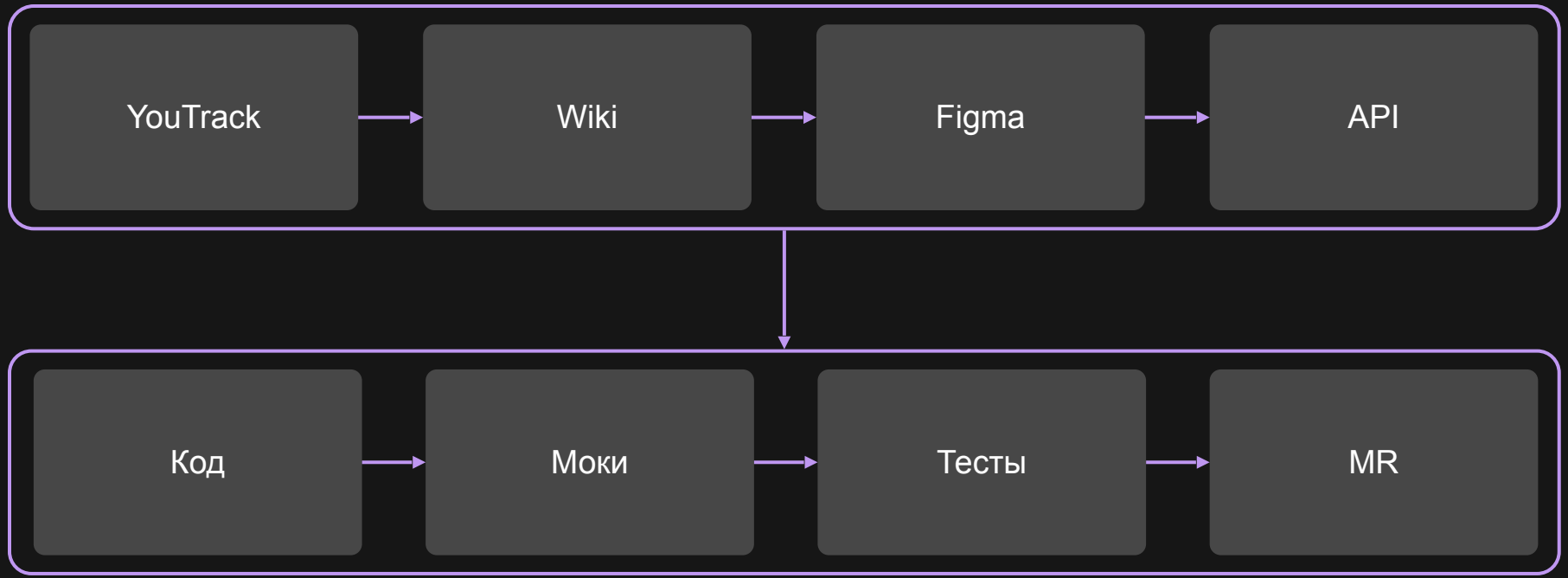
8 шагов · 8 контекстных переключений

Ручной процесс



8 шагов · 8 контекстных переключений

Ручной процесс



8 шагов · 8 контекстных переключений

Ручной процесс

Каждый переход

Переключение контекста

Каждый

Потенциальная ошибка

Цель

Автоматизировать цепочку, не звено

Архитектура

КВ + субагенты + оркестратор



AGENTS.md — точка входа

Первый файл, который читает любой агент

Стрим, KB-файлы, субагенты, порядок

Контракт проекта с AI

Knowledge Base

Причины появления

Специфичные задачи, под нейминги

Ненужный контекст других команд

Снизить потребление токенов

Knowledge Base

У каждого стрима — изолированная KB

.ai/registry/

- common/ ← для всех стримов
  - skills/ общие навыки
  - orchestration/ оркестраторы
  - kb/ общая база знаний
- streams/{stream}/ ← изоляция стрима
  - skills/ override скиллов
  - kb/ KB стрима

Фича модули в KB

code.md

экраны, флаги, Figma, нюансы

docs.md

целевое продуктовое поведение

Принцип KB

Читается первой — до любого действия

Только нужные файлы, не весь KB

Лишний контекст = шум

Принцип КВ

Специфичность

Каждый стрим: свой путь в КВ override-скиллы

Стрим-override

Подтягиваются автоматически

Изоляция

Конвенции одной команды не попадают в другую

Субагенты: вход и контекст

Агент	Роль
issue-reader	YouTrack + ссылки
doc-reader	Wiki
read-api-contracts	Чтение контрактов из Swagger

Субагенты: работа

Агент	Роль
bug-fixer	Устранение бага
enhancer	Разработка нового функционала
ui-from-figma	Макет → WBMUIKit
write-tests	Unit-тесты по референсам
test-analyzer	Покрытие

Субагенты: качество и KB

Агент	Роль
qa-checklist	Квиз перед сдачей
kb-auditor	KB vs кодовая база
auto-test	UI-автотесты (гипотеза)

Оркестратор

Оркестратор не пишет код — координирует

Один вход — любая задача

Bug → bug-fixer

Task → enhancer

Figma → ui-from-figma

Пайплайн

00

config.yml → стрим → KB

01

issue-reader → YouTrack

02

doc-reader → Wiki

03

read-api-contracts → OpenAPI

04

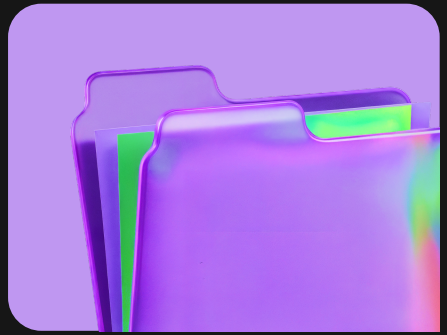
Стратегия: bug / enhancer / UI

05

test-analyzer + write-tests

06

Резюме + MR + commit message



Kilo Code

IOS-121029 issue

143,585

6 to-dos done

https://youtrack.wildberries.ru/issue/IOS-121029

I'll start by reading the task from YouTrack. Let me check the environment setup first and then read the issue.

Shell Check local agent config and .ai directory

The config is set up. Let me load the issue from YouTrack using the issue-reader subagent.

The task is a YouTrack link. Let me parse it and call the issue-reader.

Issue-Reader Agent Read YouTrack issue IOS-121029 (2)

Shell Check yt CLI availability and config

Shell Get YouTrack issue IOS-121029 as JSON

The task is a Bug about MoneyBox (Инвесткопилка) in the Dashboard (ГЭК) and Savings (Накопления) modules. The issue involves the `GOS=READY_TO_CLOSE` status and display of piggy bank balance.

Let me read the documentation link and the relevant KB modules. Let me check the domain mapping - this involves:

- Инвесткопилка, Копилка → `WBFintechMoneyBox` → `features/moneybox/code.md`
- ГЭК → `WBFintechDashboard` → `features/dashboard/code.md`
- Накопления → `WBFintechSavings` → `features/savings/code.md`

Let me read the documentation and the KB files. First, let me get the doc-reader and read the relevant KB modules.

New task

Worktree

+3484 -10

Type a message... (Enter to send, Shift+Enter for new line)

Task Orchestrator

coder-medium / coder-medium

Kilo Settings

Settings

Models

Default Model

Primary model for conversations

coder-medium / ...

Small Model

Lightweight model for ti generation, commit message generation, prompt enhancement, a other quick tasks

coder-medium / ...

Model per Mode

Code

coder-medium / ...

Add-fintech-flags

coder-medium / ...

Ask

coder-medium / ...

Auto-test

coder-medium / ...

Bugbot

Not set (use ser...

Bugfix-orchestrator

Ключевые приёмы

Что делает систему рабочей



Исходные данные — что уже было

Развитый процесс разработки макетов по дизайн-системе

Метки классов кода прямо в Figma-компонентах

Общие UI-компоненты и токены (типографика, цвета, spacing)

Однозначное соответствие: Figma → Swift-класс

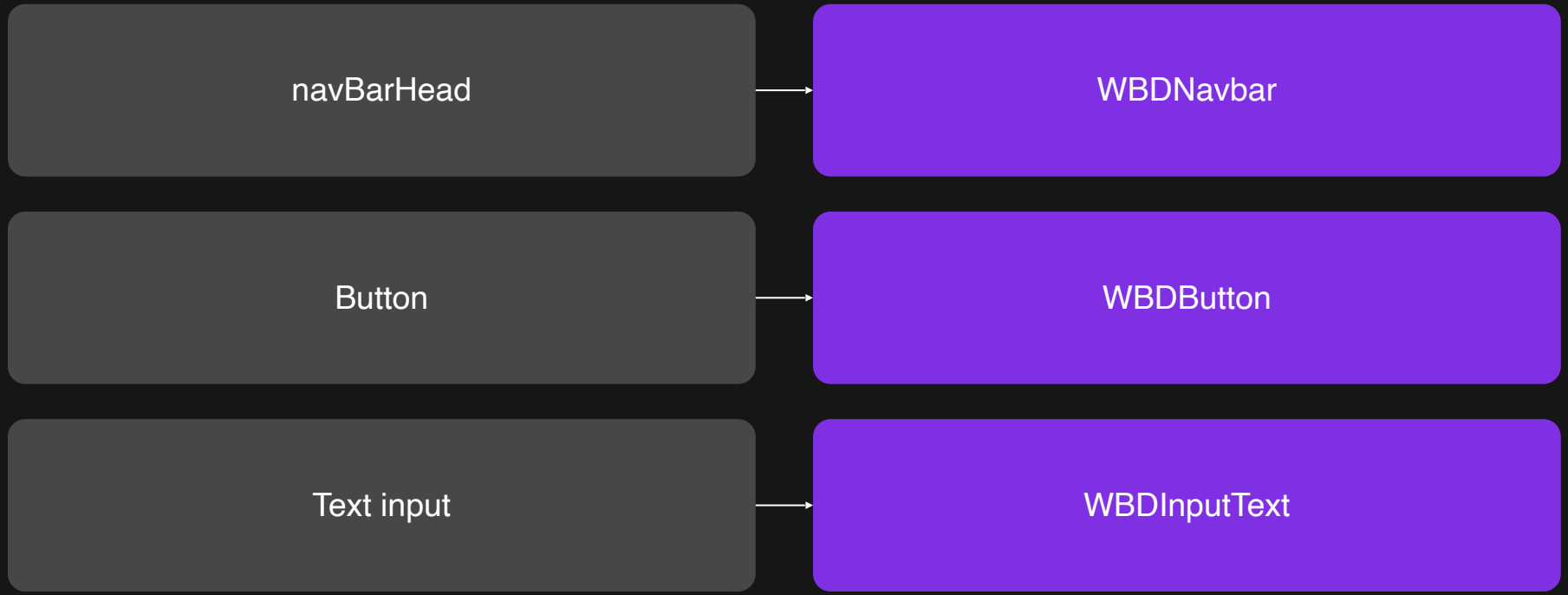
Figma через MCP

Макет читается программно

Компоненты, тексты, цвета, отступы

LLM не рисует — транслирует

Маппинг Figma → код



Контекст:
переиспользование между стримами

У каждого стрима —
свои среднеуровневые обёртки

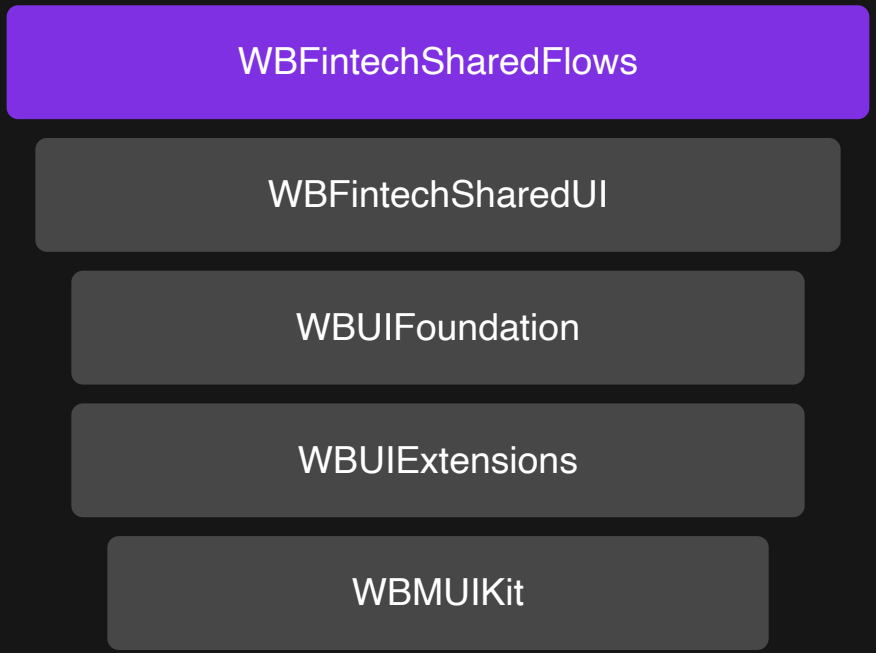
Кастомные экраны
из компонентов ДС —
переиспользуются

Разные команды
пользуются ими —
но в Figma ДС не помечены

Узнать о готовом решении —
сложно

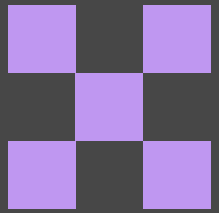
5 уровней переиспользования

Сначала ищем готовое — потом генерируем

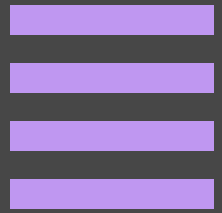


API-контракты

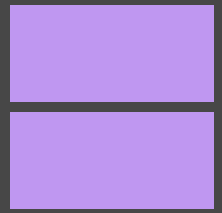
OpenAPI
из GitLab →
выжимка
в .ai/contracts/



Сверка
с endpoint-struct'ами
в коде



Сырец
удаляется —
остаётся кеш



.ai/config.yml

default_stream

Стрим, правила которого загружаем в контекст

confirm_plan

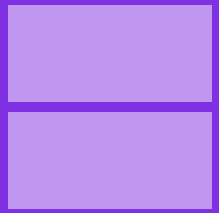
Спросить или решить самому

skip_test_analysis · ask_for_contracts

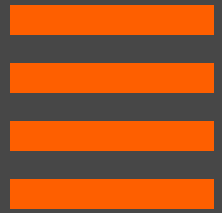
Ручки без правки скиллов

Установка в одну команду

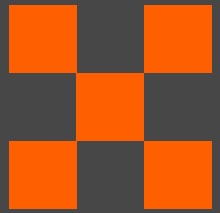
setup-ai-config.sh



Конфиг, yt, wiki, Figma, GitLab, permissions



Пока сложно — пользуются единицы



```
➜  marketplace git:(bugfix/IOS-120909) ✗
```

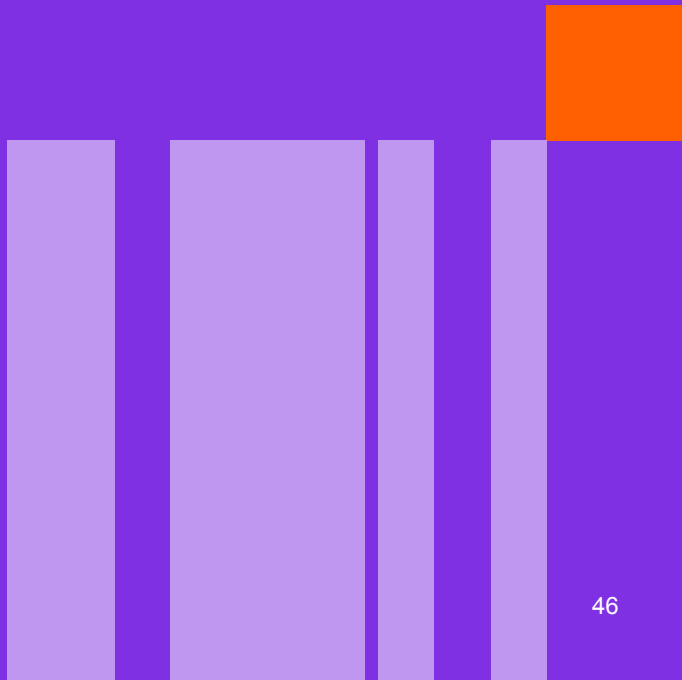
Метрики

9 месяцев · реальные задачи



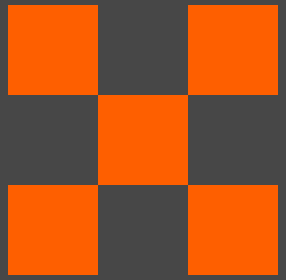
+25–100%

скорость разработки

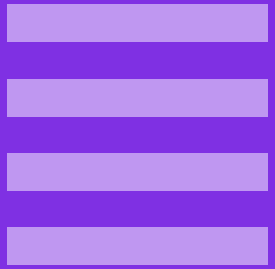


Маленькие задачи → до 100%

Сам фикс — минуты,
цикл — полчаса

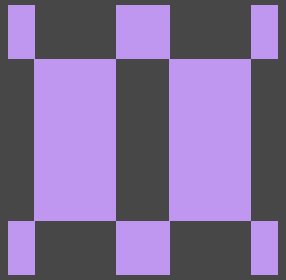


Выигрыш —
в накладных расходах,
не в «гениальности» LLM

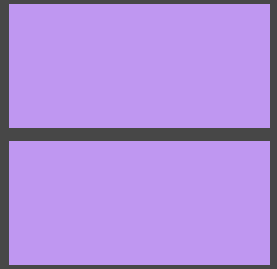


Большие задачи → ~25%

Время:
написание → проверка



Читать и поправить
быстрее, чем писать



Баги не уменьшились

Быстрее — не хуже



Неожиданная метрика

Беклог просто исчез

Неожиданная метрика

Техдолг
и минорные
фиксы —
за день

То, что лежало
неделями

AI лучше там,
где лень —
не где сложно

Ирония

Новое бутылочное горлышко



Хотели снять нагрузку с QA

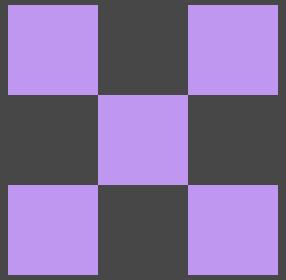
Разработчик стал быстрее

Задач и фич — больше

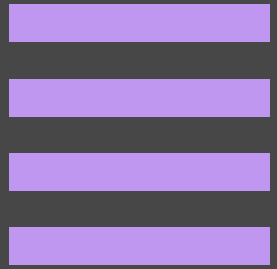
QA получила больше работы

Гипотеза: auto-test

UI-автотесты
по тест-кейсу



Ещё тестируем —
метрик нет



Автоматизация — не проект, а процесс

Ускорил одно звено — следующее стало узким местом

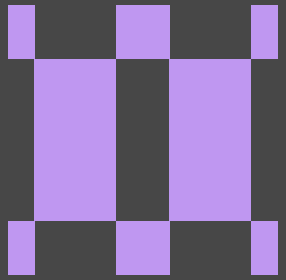
Проблемы

Метрики ≠ массовое внедрение

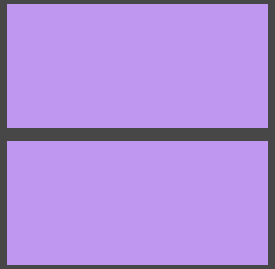


Инерция команды

Пока лиды
не пользуются —
команда тоже нет



Это не скепсис —
это инерция процесса



Вайб-кодинг

Код выглядит правильно

Логика — не та

qa-checklist + проверка по шагам

Результат борьбы

~1 из 4–5 MR всё равно уходит на доработку

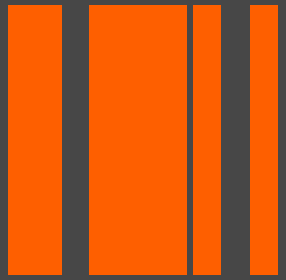
Вайб-кодинг — не проблема LLM

Проблема человека, который сканирует, а не читает

Ревью должно измениться

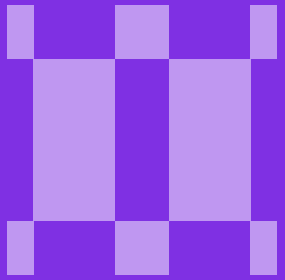
Меньше смысла

Стилистика, нейминг



Больше смысла

Логика vs ТЗ



LLM расслабляет

Сервер лежит / токен протух

Как навигатор: забыл дорогу

Новая зависимость — осознавать цену

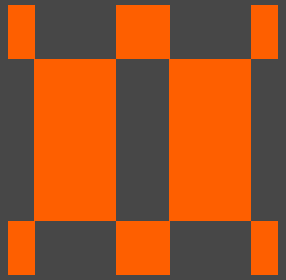
Выводы

Что вынести

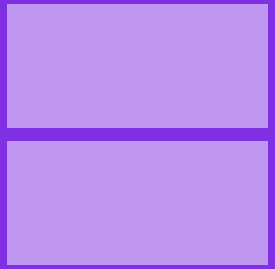


Архитектура важнее модели

Не самая
топовая LLM



KB + роли + workflow
компенсируют веса



Главный эффект — беклог

AI лучше там, где лень — не где сложно

Архитектура и оркестрация агентов

Баги не уменьшились — и это ок

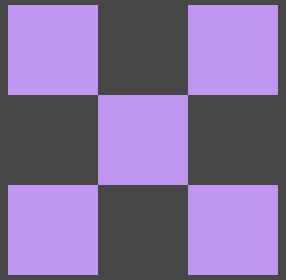
Мы стали быстрее, но не стали безошибочнее.
LLM — это усилитель, а не страховка.
Она не заменяет внимательность, она экономит время. Если вы ждёте, что AI снизит баги — вы ждёте не того

Ревью эволюционирует

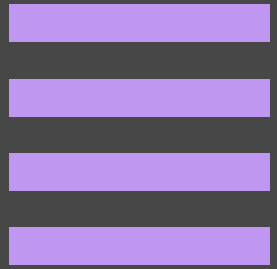
Красивый код ≠ правильный код

Риск — атрофия

Не галлюцинации — зависимость



Не запрещать — осознавать

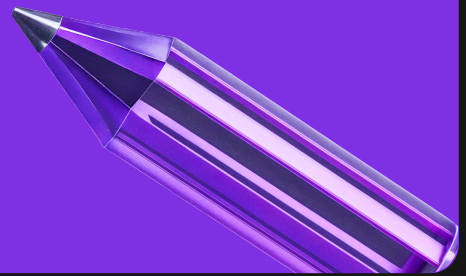


Внедрение — не техзадача

Установка —
одна команда

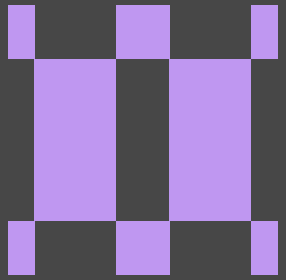


Изменить процесс —
сложно

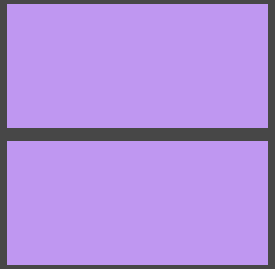


Не замена разработчику

Ревьюер, архитектор, владелец контекста

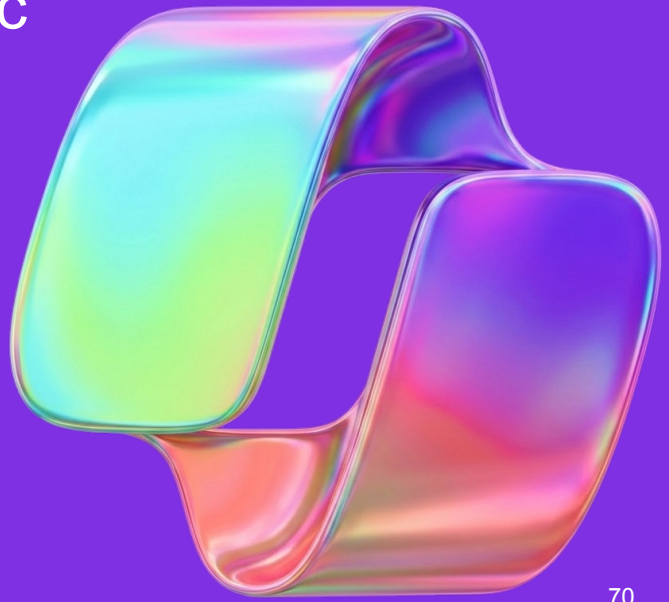


Рутину — оркестратор



Архитектура контекста и оркестрация агентов

Строим детерминированный AI-процесс



Спасибо!
Есть вопросы?



Telegram

Дмитрий Путьков
Руководитель iOS команды Инвестпродукты