




# О себе


- Team Lead в онлайн — кинотеатре КИОН
- В разработке 7 лет
- Наша команда сейчас находится в процессе перехода на использование ИИ инструментов, в частности Claude Code




# Что вы узнаете из воркшопа


 Как внедрить работу с Claude Code в ваш проект


# Что вы узнаете из воркшопа


 Как внедрить работу с Claude Code в ваш проект

 Основной функционал который предоставляет Claude Code из коробки


# Что вы узнаете из воркшопа


 Как внедрить работу с Claude Code в ваш проект


 Основной функционал который предоставляет Claude Code из коробки


 Какие могут возникать проблемы и как их можно решать

# Что вы узнаете из воркшопа

 Как внедрить работу с Claude Code в ваш проект

 Основной функционал который предоставляет Claude Code из коробки

 Какие могут возникать проблемы и как их можно решать

 Покажу что внедрение ИИ не выглядит сложной задачей

# Кому будет полезен

- Тем кто только начинает знакомиться с ИИ-агентами

# Кому будет полезен

→ Тем кто только начинает знакомиться с ИИ-агентами

→ Кто хочет на практике понять с чего начать внедрение ИИ в ваш проект

# Исходный проект




**С чем вы сразу столкнетесь ?**

- **Read**(/Users/glebkalinicenko/AndroidProjects/StroitelApp/auth-impl/src/main/java/com/itapp/auth\_impl/di/AuthGraph.kt)  
└ Read **49** lines
- **Read**(/Users/glebkalinicenko/AndroidProjects/StroitelApp/composeApp/src/main/java/com/itapp/stroitelapp/di/AppGraph.kt)  
└ Read **15** lines
- **Read**(/Users/glebkalinicenko/AndroidProjects/StroitelApp/composeApp/src/main/java/com/itapp/stroitelapp/root/RootComponent.kt)  
└ Read **14** lines
- **Read**(/Users/glebkalinicenko/AndroidProjects/StroitelApp/composeApp/src/main/java/com/itapp/stroitelapp/root/RootComponentImpl.kt)  
└ Read **60** lines
- **Read**(/Users/glebkalinicenko/AndroidProjects/StroitelApp/composeApp/src/main/java/com/itapp/stroitelapp/App.kt)  
└ Read **24** lines


# Проблема


 LLM начинает собирать полный контекст проекта

# Следствие

 Расширять исходный промпт дополнительной “общей” информацией;

# Следствие

 Расширять исходный промпт дополнительной "общей" информацией;

 Неконсистентность "общей информации";

# Причина

Claude Code стартует каждую сессию  
с нулевым знанием о проекте

Что нам может помочь?



# Что это?

- Markdown — файл с базовым контекстом о вашем проекте

# Что это?

→ Markdown — файл с базовым контекстом о вашем проекте

→ Claude загружает его каждый при запуске новой сессии

# Содержимое Claude.md

Перед тем как описывать файл Claude.md, ответьте на вопросы:

# Содержимое Claude.md

Перед тем как описывать файл Claude.md, ответьте на вопросы:

 О чем ваш проект;

# Содержимое Claude.md

Перед тем как описывать файл Claude.md, ответьте на вопросы:



О чем ваш проект;



Какие технологии используются на проекте;

# Содержимое Claude.md

Перед тем как описывать файл Claude.md, ответьте на вопросы:



О чем ваш проект;



Какие технологии используются на проекте;



Как ИИ может взаимодействовать с вашим проектом;

# Содержимое Claude.md

Перед тем как описывать файл Claude.md, ответьте на вопросы:



О чем ваш проект;



Какие технологии используются на проекте;



Как ИИ может взаимодействовать с вашим проектом;



Дополнительная общая полезная информация;

# Live coding



# Последствия

- ✓ Claude Code знает общую информацию о проекте;

# Последствия

- ✓ Claude Code знает общую информацию о проекте;
- ✓ Экономия времени на сборке контекста проекта;

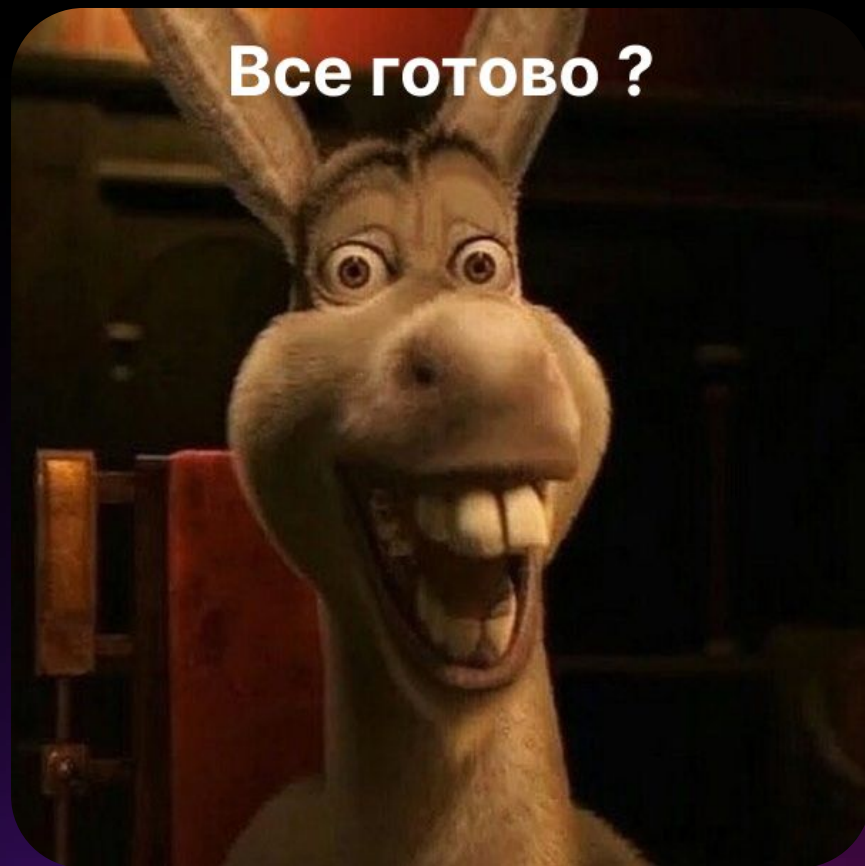
# Последствия

- ✓ Claude Code знает общую информацию о проекте;
- ✓ Экономия времени на сборке контекста проекта;
- ✓ Единое хранилище "общей конфигурацией" проекта;

# Последствия

✗ Поддержка актуальности файла;


**Все готово ?**




# Проблема


Код новых фичей, не всегда был написан  
как "принято" в команде

# Следствие


 Расширение описания “правильных” подходов в Claude.md;


# Следствие


 Расширение описания “правильных” подходов в Claude.md;

 Разрастание размера файла Claude.md и ухудшение его читаемости;

# Следствие

 Расширение описания “правильных” подходов в Claude.md;

 Разрастание размера файла Claude.md и ухудшение его читаемости;

 Периодическое игнорирование Claude.md моделью, из-за релевантного контекста;

# Причина

LLM не всегда понимает какие подходы  
"правильные" на проекте

Что нам может помочь?

# Решение



# Решение

**SKILLS** - набор инструкций, который расширяет поведение Claude Code

# Польза



Меньше ручных инструкций;

# Польза

 Больше ручных инструкций;

 Одно место для хранения "экспертизы";

# Польза

 Больше ручных инструкций;

 Одно место для хранения "экспертизы";

 Переиспользуемость;

# Польза

 Больше ручных инструкций;

 Одно место для хранения "экспертизы";

 Переиспользуемость;

 Ленивая загрузка;

# Как работают Skills



## Claude Agent

Получает задачу:

**Реализовать экран фильтров  
для фильмов и сериалов**

# Как работают Skills



## Claude Agent

Читает:

`architecture/SKILL.md`

Получает знания:


- о clean архитектуре;
- устройстве модулей;
- написанию UI-кода;
- ...



## Skills

 architecture

 compose

 unit-testing

 coroutines

# Как работают Skills



## Claude Agent

Следует инструкциям из:  
`architecture/SKILL.md`


- ✓ Консистентность
- ✓ Чистота кода
- ✓ Лучшие практики



## Skills

 architecture

 compose

 unit-testing

 coroutines

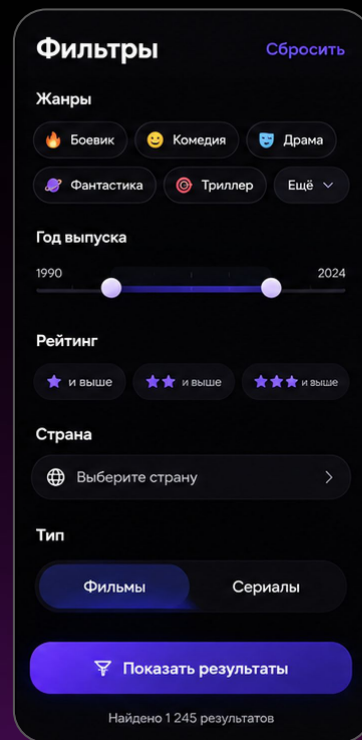
# Как работают Skills



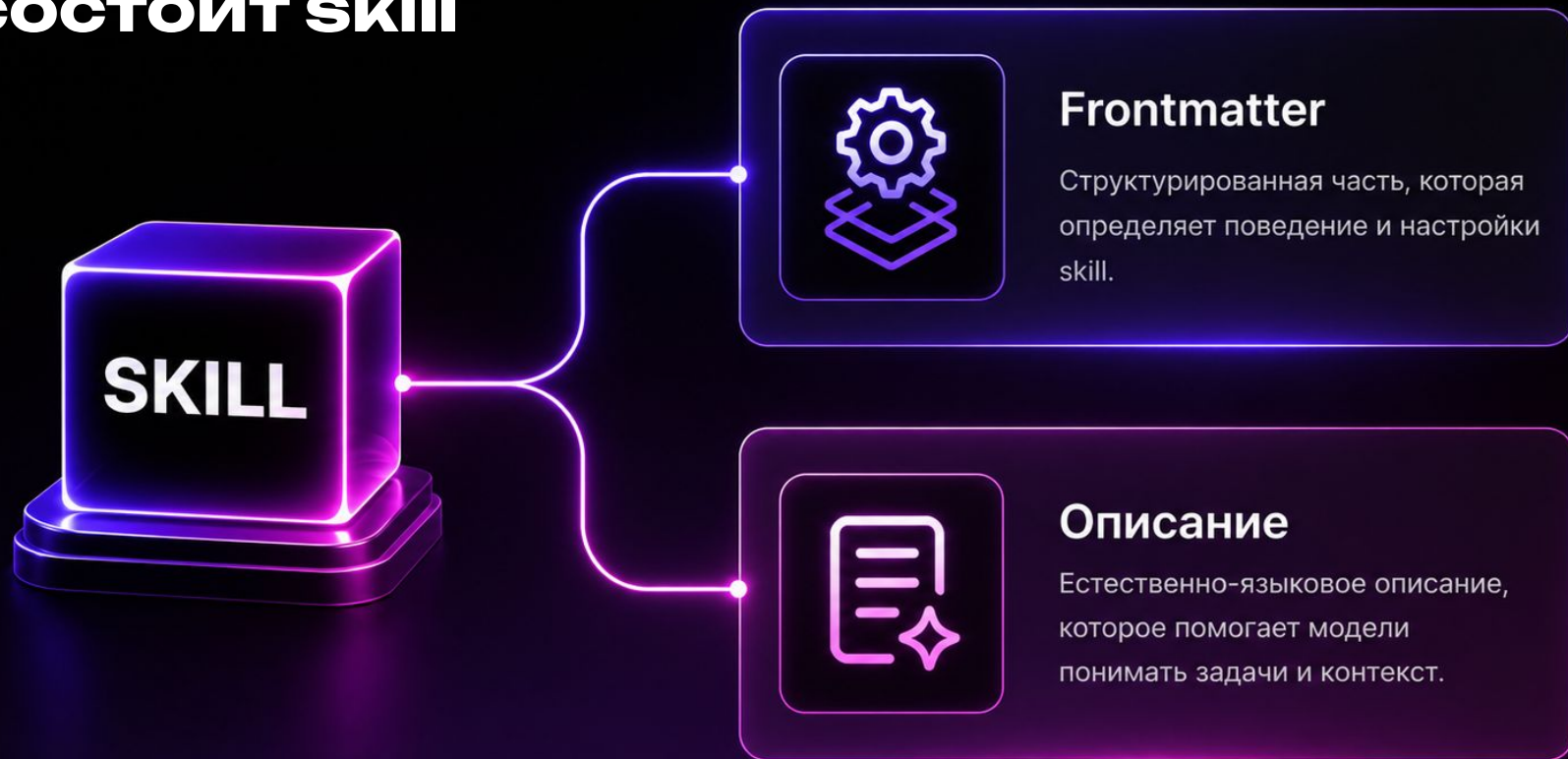
Claude Agent

Результат

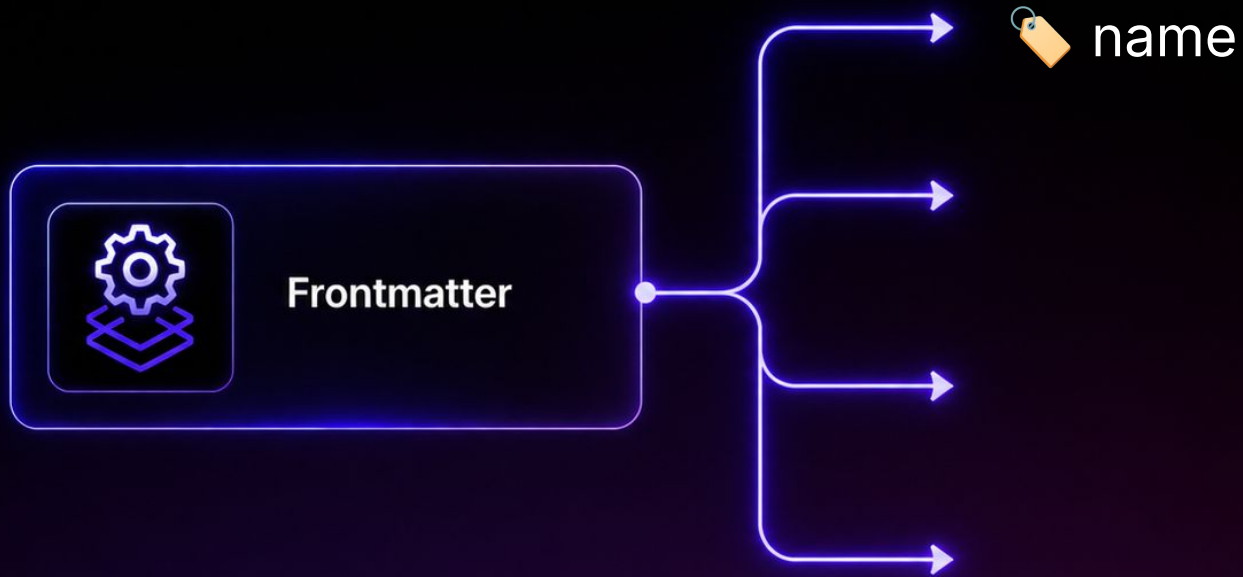
Новый экран фильтров  
для фильмов и сериалов



# Из чего состоит skill



# Структура frontmatter



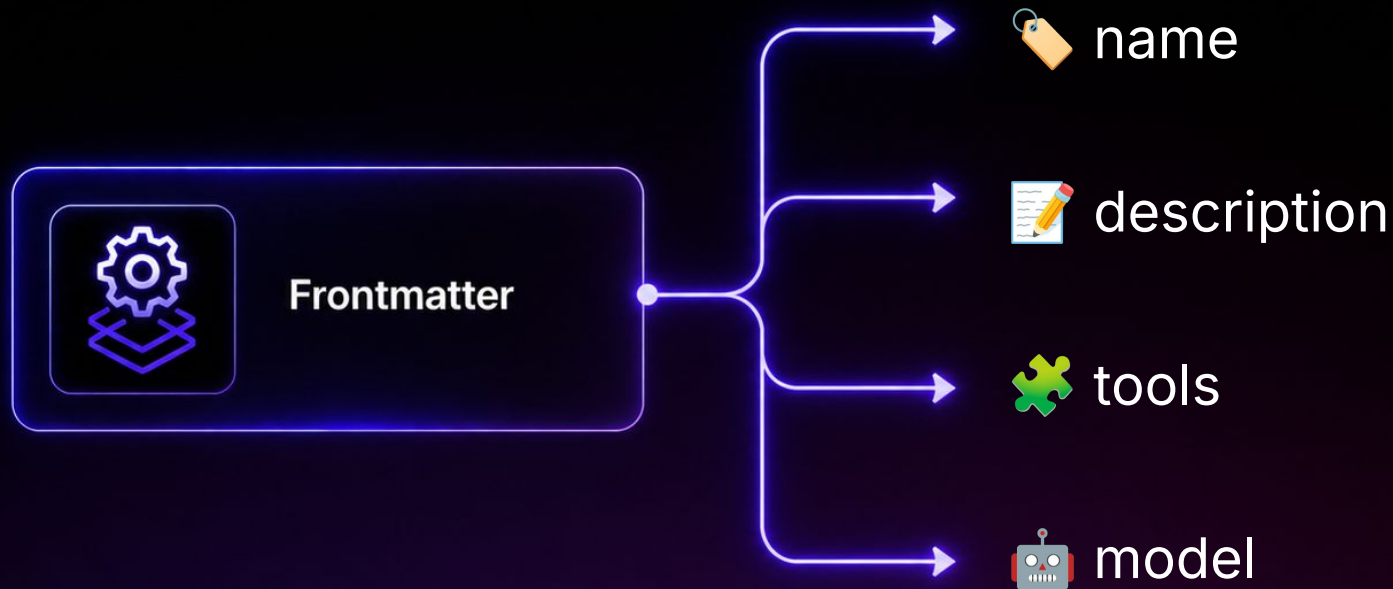
# Структура frontmatter



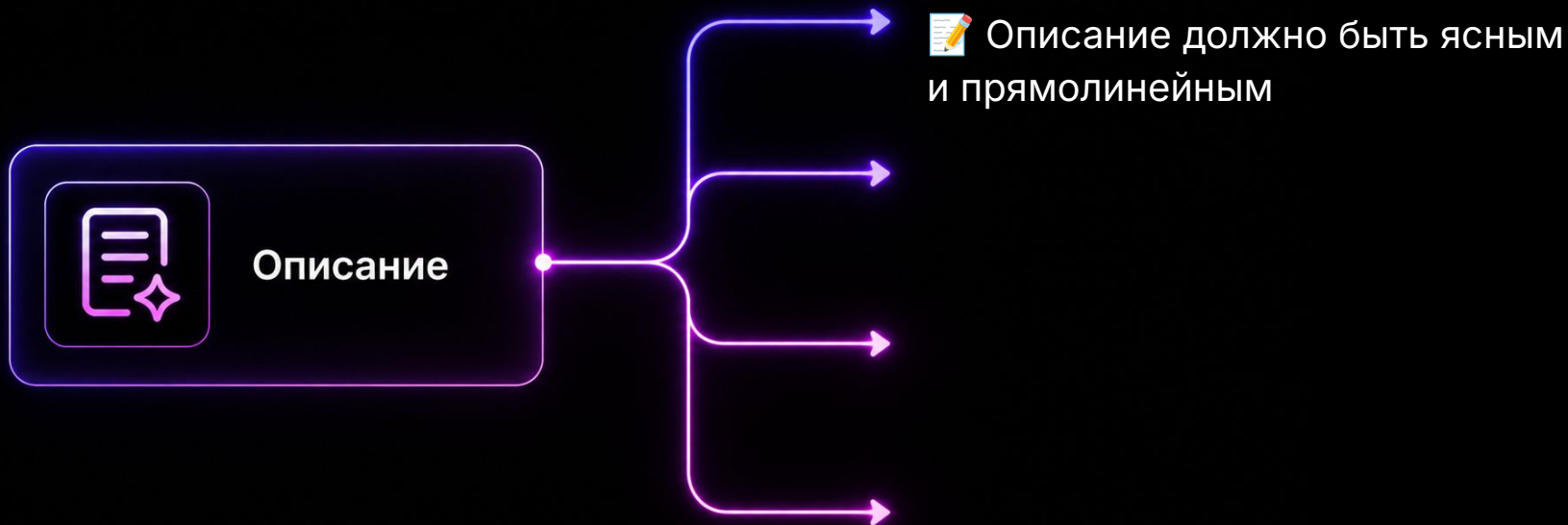
# Структура frontmatter



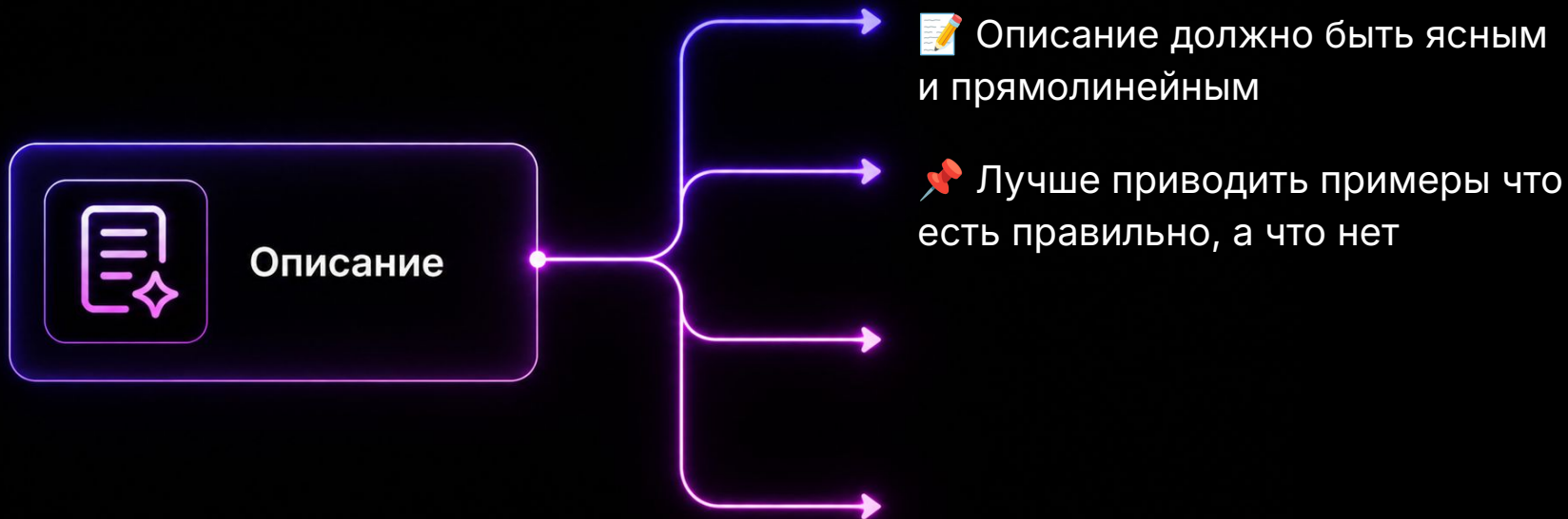
# Структура frontmatter



# Структура описания описания



# Структура описания описания



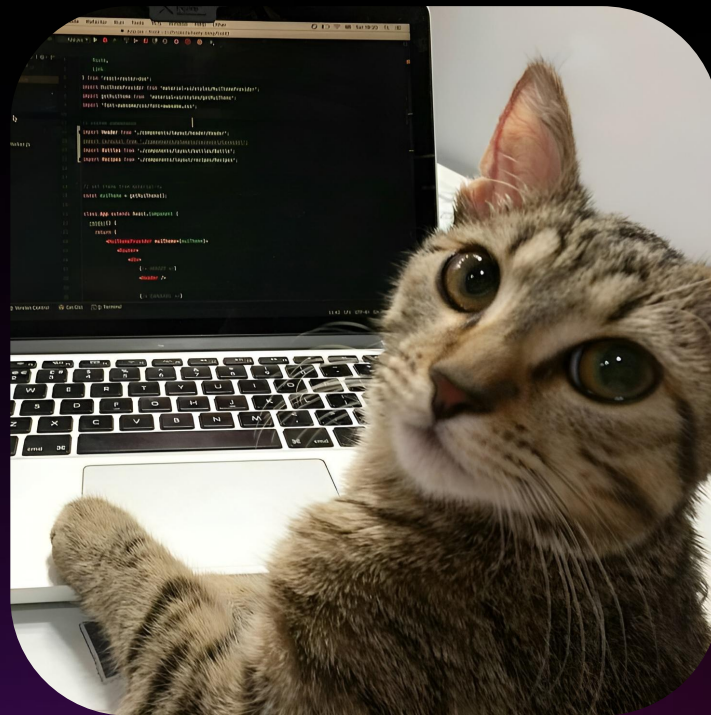
# Структура описания описания



# Структура описания описания



# Live coding



# Последствия

- ✓ Генерация нового кода, следует "конвенциям";

# Последствия

- ✓ Генерация нового кода, следует "конвенциям";
- ✓ Сокращение размера Claude.md;

# Последствия

- ✓ Генерация нового кода, следует “конвенциям”;
- ✓ Сокращение размера Claude.md;
- ✓ Знание о проекте стали явно задокументированы;

# Последствия

- ✓ Генерация нового кода, следует “конвенциям”;
- ✓ Сокращение размера Claude.md;
- ✓ Знание о проекте стали явно задокументированы;
- ✓ Стало меньше галлюцинаций;

# Последствия

✗ Поддержка актуальности файла;

# Последствия

- ✗ Поддержка актуальности файла;
- ✗ Важно следит за описанием skill;

**А теперь все готово ?**



# Проблема

Деградация контекста при длительных сессиях

# Следствие

Итеративная корректировка;

# Следствие

✗ Итеративная корректировка;

# Следствие

✗ Итеративная корректировка;

Новая сессия;

# Следствие

✗ Итеративная корректировка;

✗ Новая сессия;

# Следствие

✗ Итеративная корректировка;

✗ Новая сессия;

Делегировать Claude запоминание важного;

# Следствие

- ✗ Итеративная корректировка;
- ✗ Новая сессия;
- ✗ Делегировать Claude запоминание важного;

**Почему это происходит ?**

**Context is everywhere!!!**





**Контекст** — максимальное количество входящих токенов, которые модель может использовать за один раз

# Что попадает в контекст



Входные сообщения



Выходные сообщения



Skills



Сабагенты



MCP



tools

...

# Причина

- Модель галлюцинирует при росте занятого контекста;

# Причина

- Модель галлюцинирует при росте занятого контекста;
- Контекст имеет фиксированный размер и может заканчиваться;


# Исследования

- Maximum Effective Context Window
- Context Rot: How Increasing Input Tokens Impacts LLM Performance

# Что исследовали

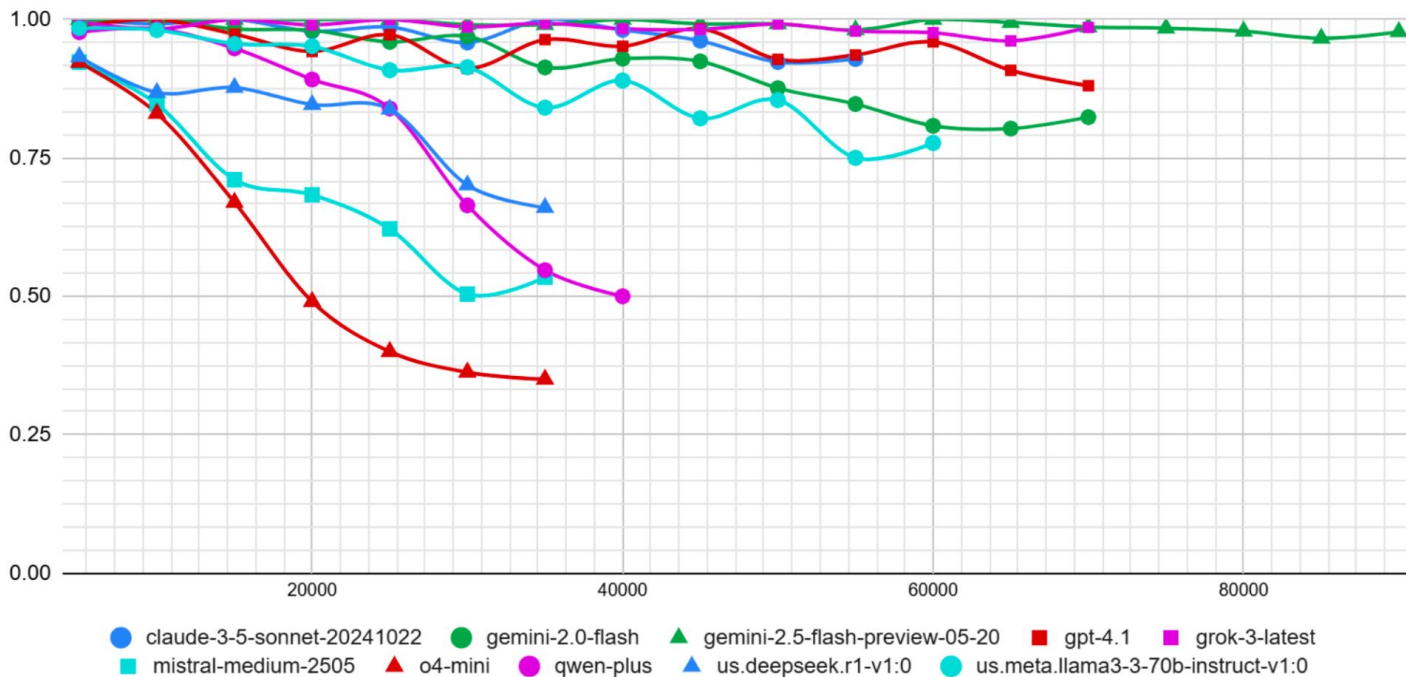
- 🧠 Исследовали реальный предел «контекста» у больших языковых моделей

# Задача Needle-in-a-Haystack

 Задача — в длинном тексте спрятан один нужный факт, нужно найти его в контексте

# Задача Needle-in-a-Haystack

Needle in a Haystack




# Ключевые наблюдения

 Качество падает с ростом контекста;

 Ошибки могут появляться уже на сотнях токенов;

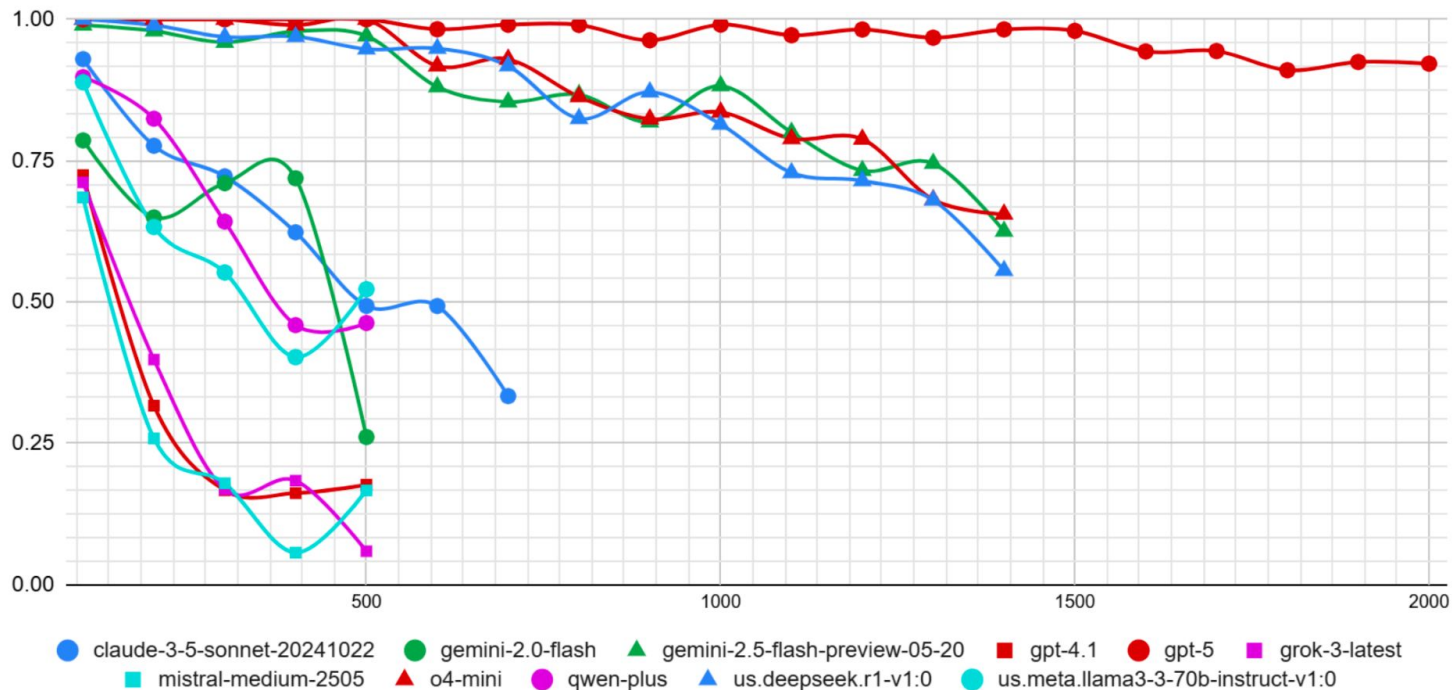
 Поведение нестабильное (резкие провалы);

# Задача Find & Sort

 Задача — в длинном тексте нужно найти несколько элементов, отфильтровать лишнее и правильно их упорядочить

# Задача Find & Sort

Sort Question




# Ключевые наблюдения

⚠ Деградация начинается раньше, чем в простом поиске;

💡 Ошибки накапливаются между шагами;

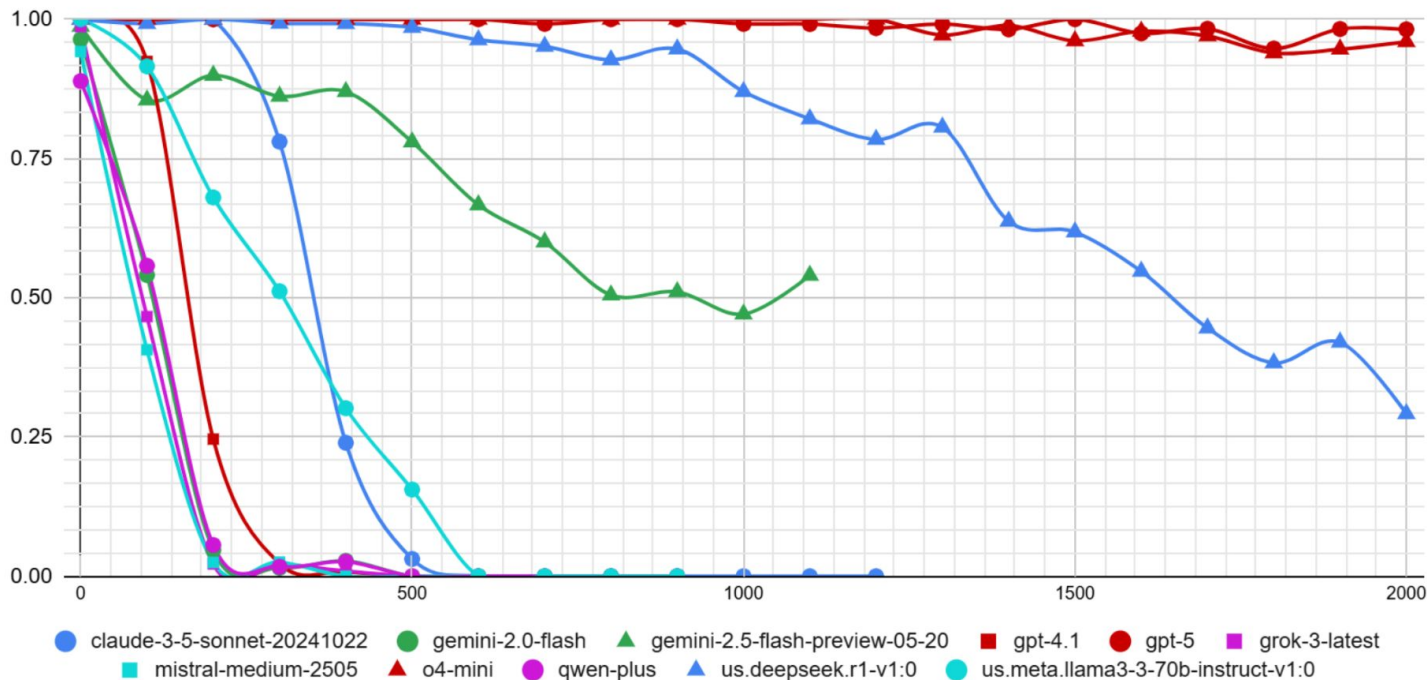
📉 Качество падает быстрее и сильнее;

# Задача Summarization

 Задача — нужно обработать весь текст целиком, выделить главное, сохранить смысл

# Задача Summarization

Summary Question



# Ключевые наблюдения

- ⚠️ Сильная деградация при росте контекста;
- 🔬 Модель "перегружается" объёмом информации;
- 📊 Часто хуже, чем задачи с выбором данных;

# Обобщенный вывод

▲ Модели деградируют намного раньше, чем заявленный лимит от вендора;

# Обобщенный вывод

⚠ Модели деградируют намного раньше, чем заявленный лимит от вендора;

💡 Деградация контекста может быть заметна и при ~100 токенов;

# Обобщенный вывод

⚠ Модели деградируют намного раньше, чем заявленный лимит от вендора;

💡 Деградация контекста может быть заметна и при ~100 токенов;

🧠 У большинства моделей заметная деградация к ~1000 токенам;

**Что нам может помочь?**

# Решение



Сабагенты

# Что это?

**Субагент** — отдельная конфигурация Claude, которая может быть вызвана главным агентом для решения определенной задачи



# Схема работы

## Main Agent

Контекстное окно

История диалога

Код проекта

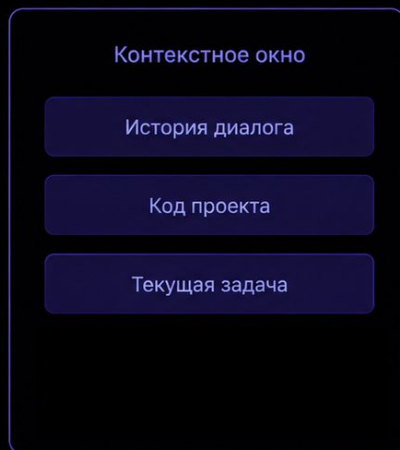
Текущая задача



Полный доступ к Tools  
read · write · bash

# Схема работы

## Main Agent



Полный доступ к Tools  
read · write · bash

① Подзадача



## Subagent

Изолированный контекст

Только подзадача

Релевантные файлы

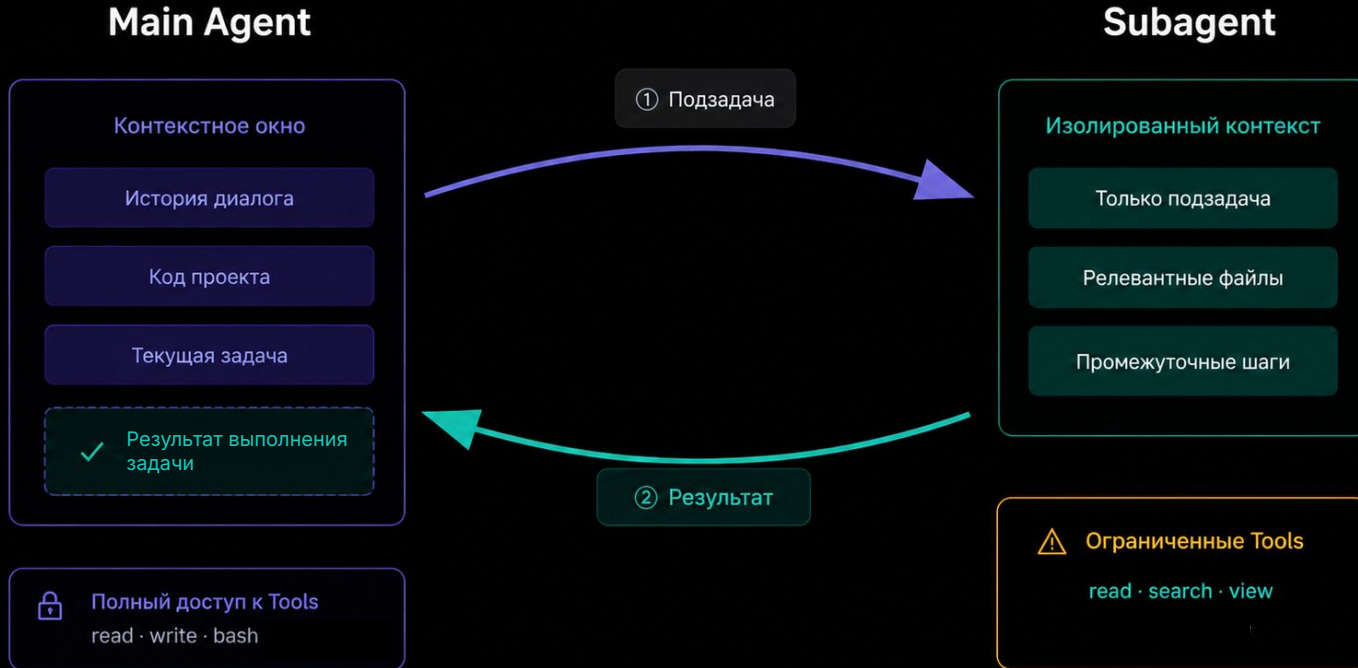
Промежуточные шаги



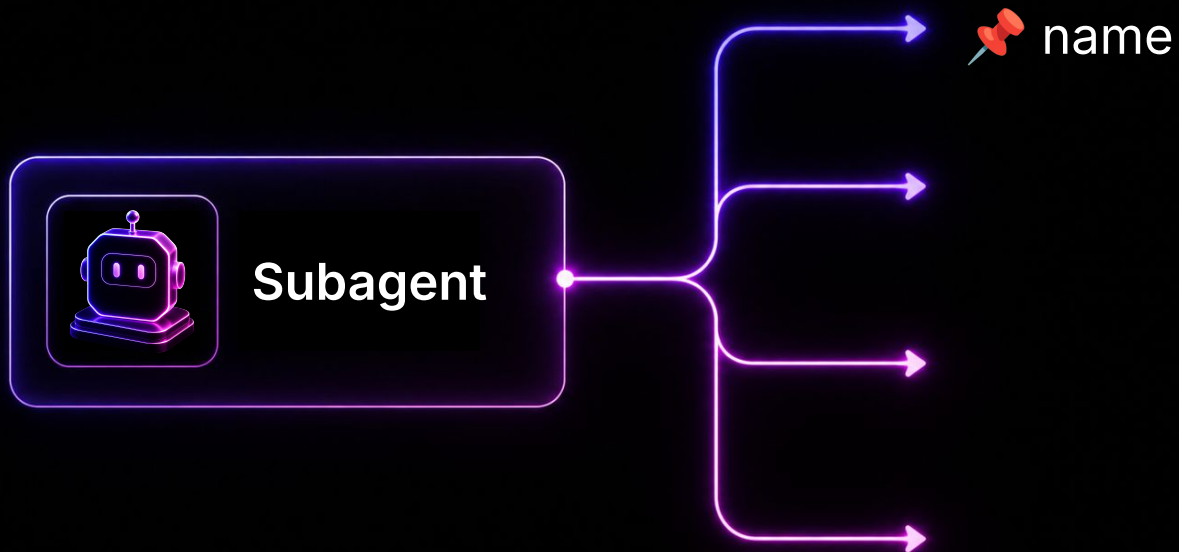
Ограниченные Tools

read · search · view

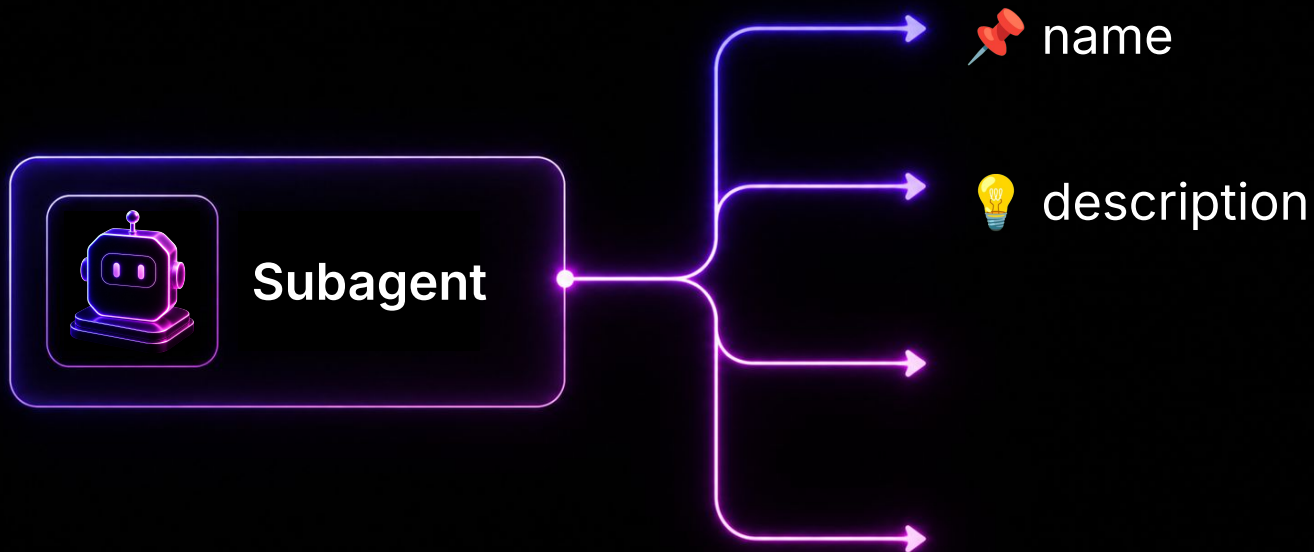
# Схема работы



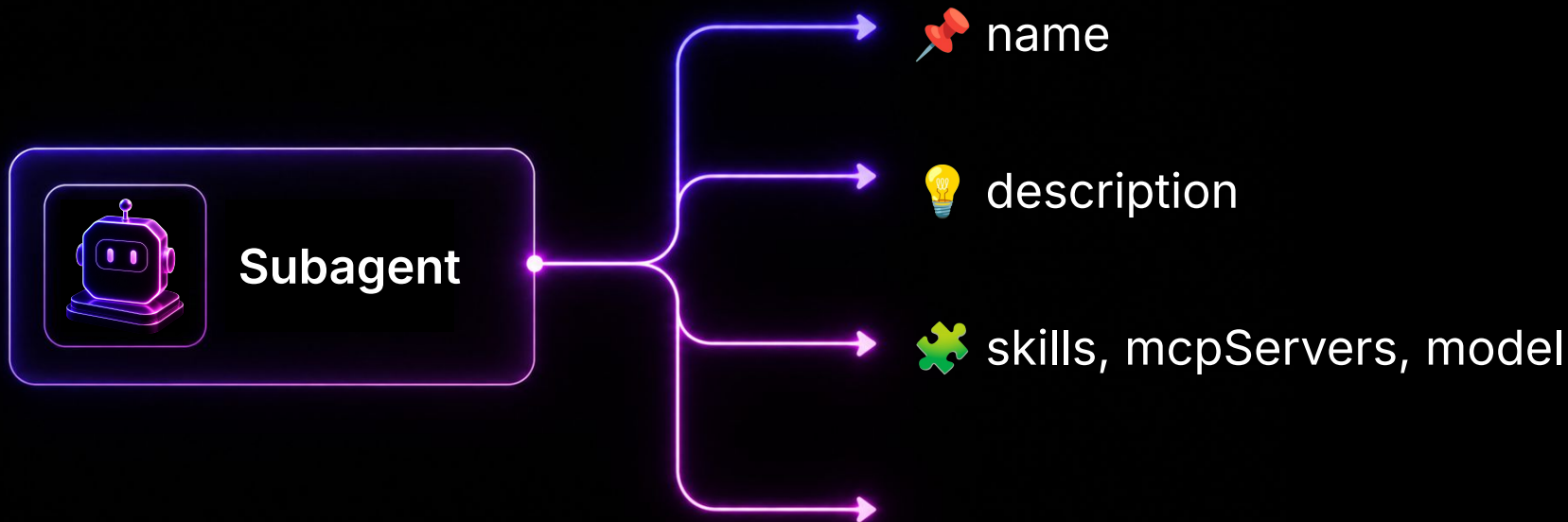
# Структура subagent



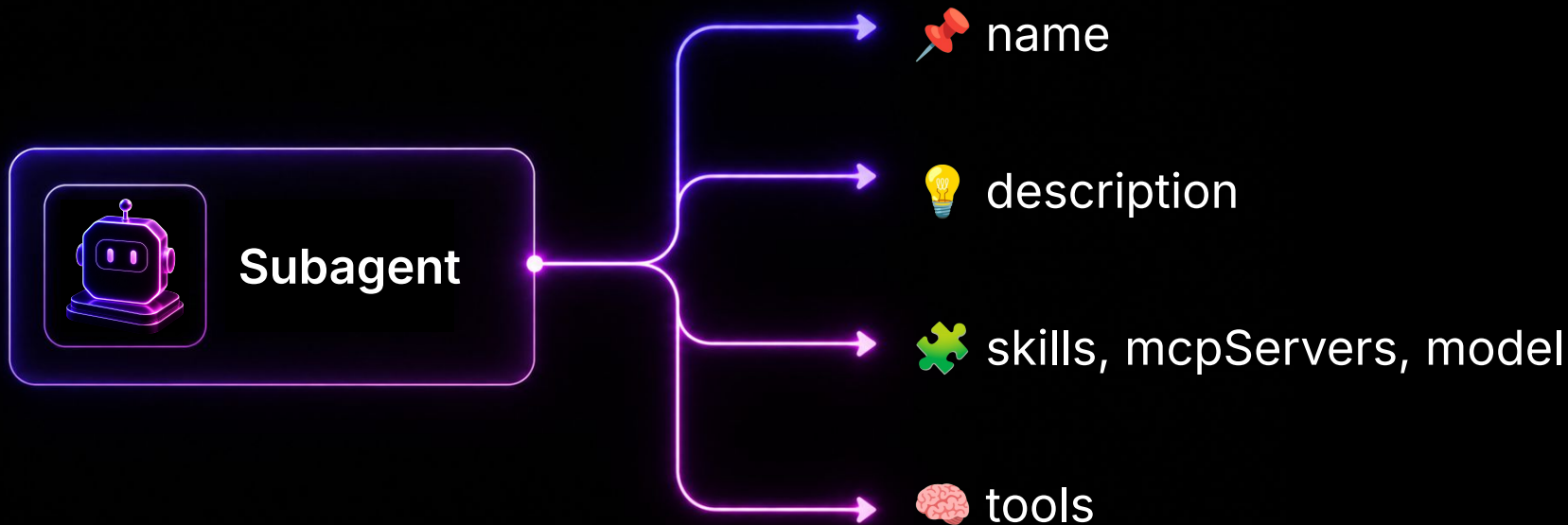
# Структура subagent



# Структура subagent



# Структура subagent



# Признаки сабагента

→ Выполняет одну ответственность;

# Признаки сабагента

- Выполняет одну ответственность;
- Может работать независимо или параллельно основному агенту;

# Признаки сабагента

- Выполняет одну ответственность;
- Может работать независимо или параллельно основному агенту;
- Может работать с отдельными набором инструментов;

# Признаки сабагента

- Выполняет одну ответственность;
- Может работать независимо или параллельно основному агенту;
- Может работать с отдельными набором инструментов;
- Может работать с конкретной моделью;

# Потенциальные сабагенты

→ Агент для исследования задачи

# Потенциальные сабагенты

- Агент для исследования задачи;
- Агент для code review;

# Потенциальные сабагенты

- Агент для исследования задачи;
- Агент для code review;
- Агент для написания документации;

# Потенциальные сабагенты

- Агент для исследования задачи;
- Агент для code review;
- Агент для написания документации;
- Агент для тестирования фичи;

# Live coding



# Последствия

✓ Меньше загрузка контекста основного агента;

# Последствия

- ✓ Меньше загрузка контекста основного агента;
- ✓ Уменьшение время работы Claude за счет параллельного выполнения;

# Последствия

- ✓ Меньше загрузка контекста основного агента;
- ✓ Уменьшение время работы Claude за счет параллельного выполнения;
- ✗ Поддержка актуальности;

# Последствия

- ✓ Меньше загрузка контекста основного агента;
- ✓ Уменьшение время работы Claude за счет параллельного выполнения;
- ✗ Поддержка актуальности;
- ✗ Высокое качество описания для активации агента;

# Последствия

- ✓ Меньше загрузка контекста основного агента;
- ✓ Уменьшение время работы Claude за счет параллельного выполнения;
- ✗ Поддержка актуальности;
- ✗ Высокое качество описания для активации агента;
- ✗ Количество потребляемых токенов увеличилось;


**Может ли нам еще что-то помочь?**

# Решение



**Compact**

# Compact

 Команда предназначенная для сжатия истории переписки, с целью освободить место в контекстном окне модели.

# До суммаризации контекста

## Context Usage

☉ ☉

☉ ☉

☐ ☐

☐ ☐

☐ ☐

☐ ☐

(0.9%)

☐ ☐

(1.6%)

☐ ☐

(0.2%)

☐ ☐

(0.5%)

☐ ☐

(3.3%)

Opus 4.7 (1M context)

claude-opus-4-7[1m]

118.9k/1m tokens (12%)

*Estimated usage by category*

☉ System prompt: 8.9k tokens

☉ System tools: 16.4k tokens

☉ Custom agents: 1.6k tokens

☉ Memory files: 4.8k tokens

☉ Skills: 1.1k tokens (0.1%)

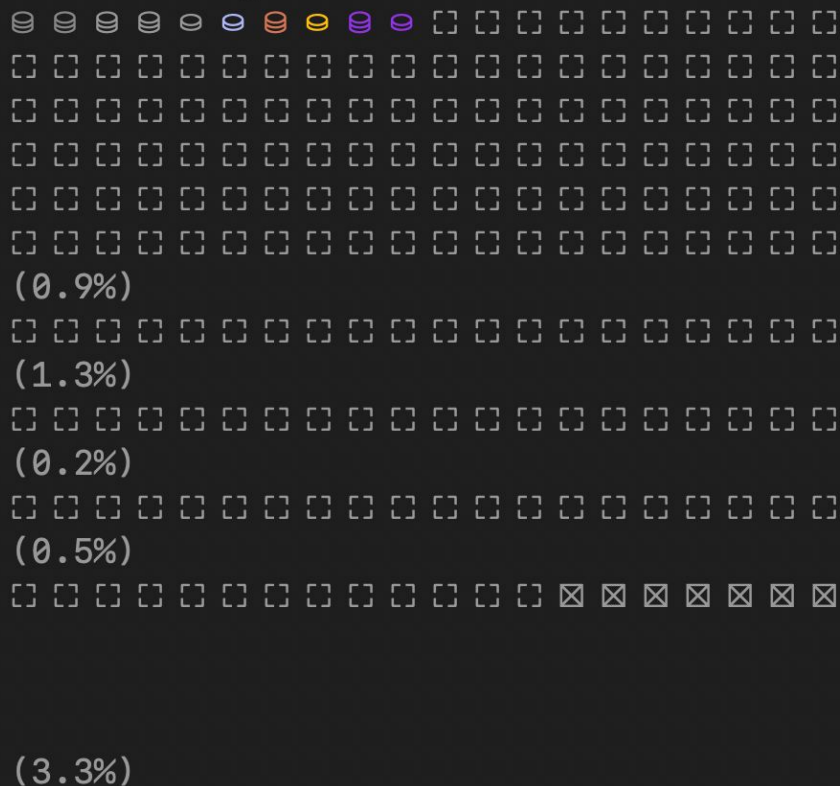
☉ Messages: 92.6k tokens (9.3%)

☐ Free space: 841.7k (84.2%)

☒ Autocompact buffer: 33k tokens

# После суммаризации контекста

## Context Usage



Opus 4.7 (1M context)

claude-opus-4-7[1m]

37.4k/1m tokens (4%)

*Estimated usage by category*

☉ System prompt: 8.9k tokens

☉ System tools: 12.7k tokens

☉ Custom agents: 1.6k tokens

☉ Memory files: 4.8k tokens

☉ Skills: 1.1k tokens (0.1%)

☉ Messages: 8.4k tokens (0.8%)

☐ Free space: 929.6k (93.0%)

☒ Autocompact buffer: 33k tokens

# Последствия

- ✓ Сжимает предыдущие сообщения в краткие резюме;

# Последствия

- ✓ Сжимает предыдущие сообщения в краткие резюме;
- ✓ Освобождает токены для новых задач и файлов;

# Последствия

- ✓ Сжимает предыдущие сообщения в краткие резюме
- ✓ Освобождает токены для новых задач и файлов
- ✓ Пытается сохранить ключевые факты при суммаризации;

# Последствия

✗ Нет контроля над тем, что будет сохранено;

# Последствия


- ✗ Нет контроля над тем, что будет сохранено;
- ✗ Потенциальная потеря важных данных в контексте;

# Последствия

- ✗ Нет контроля над тем, что будет сохранено;
- ✗ Потенциальная потеря важных данных в контексте;
- ✗ Не получится переиспользовать результат в новой сессии;

**А еще варианты ?**

# State compression

 В промежуточной точке можно сохранить текущее состояние работы с Claude Code

# Структура state файла



Цель

# Структура state файла

 Цель

 Текущий план

# Структура state файла



Цель



Текущий план



Текущие факты

# Структура state файла



Цель



Текущий план



Текущие факты



Релевантные файлы

# Структура state файла



Цель



Текущий план



Текущие факты



Релевантные файлы



Текущие решения

# Структура state файла



Цель



Текущий план



Текущие факты



Релевантные файлы



Текущие решения



Риски

# Структура state файла



Цель



Текущий план



Текущие факты



Релевантные файлы



Текущие решения



Риски



Следующие шаги

# State файл

## Main Agent

① Анализ постановки задачи  
разработки экрана фильтров

② Проектирование технического  
плана реализации

③ Разработка data-слоя  
приложения

④ Разработка domain-слоя  
приложения

# State файл

## Main Agent

① Анализ постановки задачи  
разработки экрана фильтров

② Проектирование технического  
плана реализации

③ Разработка data-слоя  
приложения

④ Разработка domain-слоя  
приложения



State.md

### Фича

Разработка экрана фильтров

### Сделано

- DataSource, Dto модели
- Repository, Mapper
- UseCase

### Решения

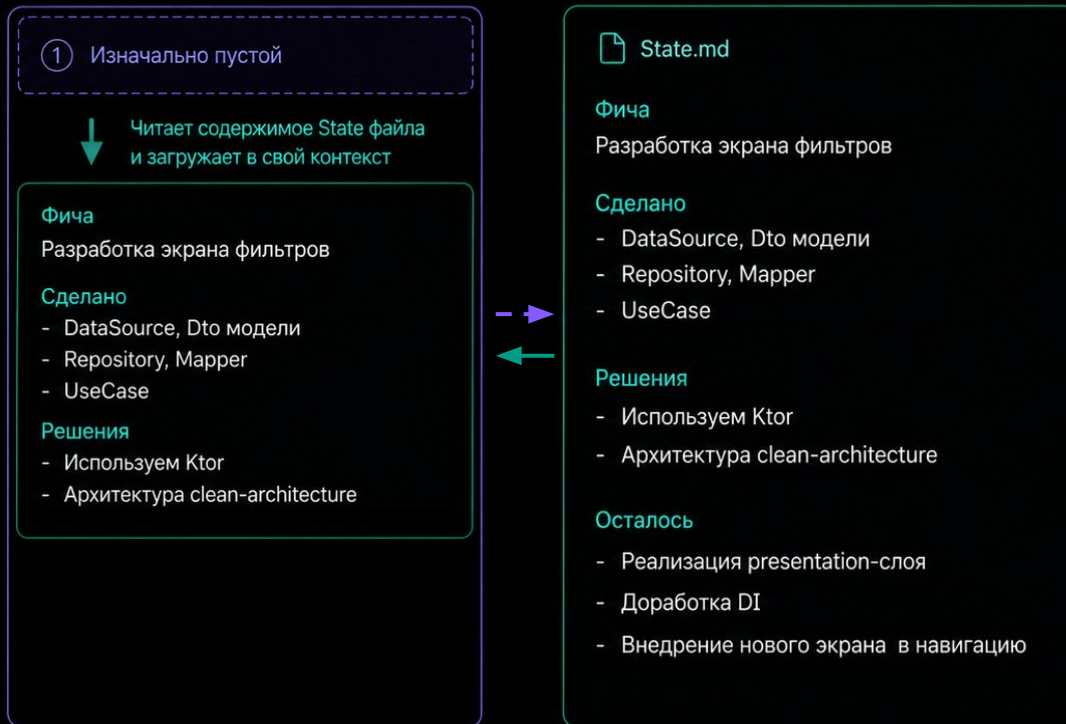
- Используем Ktor
- Архитектура clean-architecture

### Осталось

- Реализация presentation-слоя
- Доработка DI
- Внедрение нового экрана в навигацию

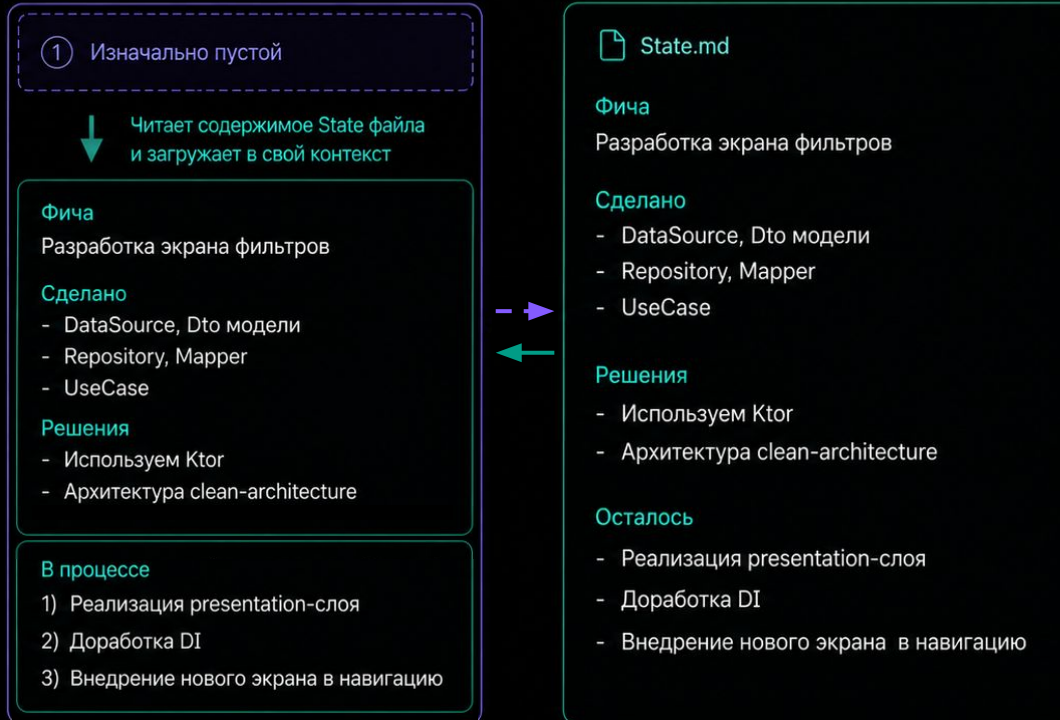
# State файл

## Main Agent



# State файл

## Main Agent



# Последствия

- ✓ Упрощение работы с долгими задачами;

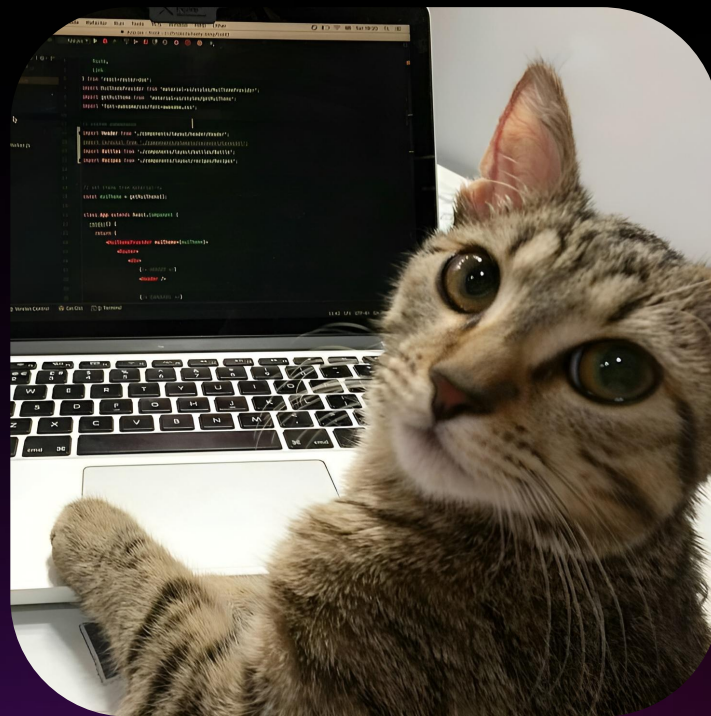
# Последствия

- ✓ Упрощение работы с долгими задачами;
- ✓ Возможность валидации "Промежуточного" файла;

# Последствия

✗ Поддержка актуальности;

# Live coding фичи



**А теперь все готово ?**



# Проблема

“На средних задачах результат не совпадал с тем, что я имел в виду”

# Следствие

✗ Попытки делать промпт более длинным и детальным;

# Следствие

- ✗ Попытки делать промпт более длинным и детальным;
- ✗ Интерактивный подход к реализации — промпт за промптом;


# Причина

 Модель не знает про неявный контекст у нас в голове;


# Причина

- 🧠 Модель не знает про неявный контекст у нас в голове;
- 📌 Введенный промпт не всегда является финальной спецификацией задачи;

# Решение




 Вынести формирование постановки задачи в отдельный документ — спеку;

# Решение

 Вынести формирование постановки задачи в отдельный документ — спеку;

 Формирование технического плана реализации фичи на базе спеки;

# Решение

-  Вынести формирование постановки задачи в отдельный документ — спеку;
-  Формирование технического плана реализации фичи на базе спеки;
-  Реализация отдельных задач на базе технического плана;

# Решение

## 1 Требования

Диалог с Claude  
по команде /spec

-----  
Артефакт:

**spec.md**

что делает, границы,  
edge cases, не-цели

# Решение

## 1 Требования

Диалог с Claude  
по команде /спес

-----  
Артефакт:

**спес.md**

что делает, границы,  
edge cases, не-цели

на базе  
спеки



## 2 План

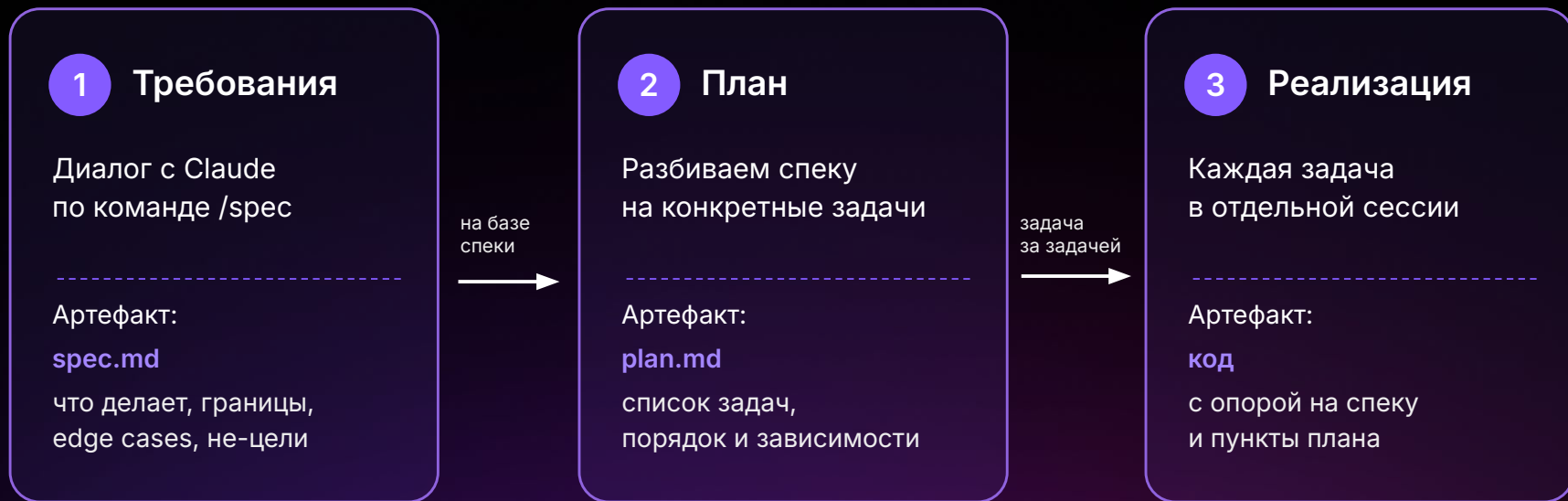
Разбиваем спеку  
на конкретные задачи

-----  
Артефакт:

**plan.md**

список задач,  
порядок и зависимости

# Решение




# Структура зрес-файла

 Контекст;


# Структура sres-файла

 Контекст;

 Цель;

# Структура sres-файла

 Контекст;

 Цель;

 Функциональные требования;

# Структура sres-файла

 Контекст;

 Цель;

 Функциональные требования;

 Нефункциональные требования;

# Структура sres-файла

 Контекст;

 Цель;

 Функциональные требования;

 Нефункциональные требования;

 Корнер-кейсы;

# Структура sres-файла

 Контекст;

 Цель;

 Функциональные требования;


 Нефункциональные требования;

 Корнер-кейсы;

 Out of scope;

# Структура sres-файла

 Контекст;

 Цель;

 Функциональные требования;

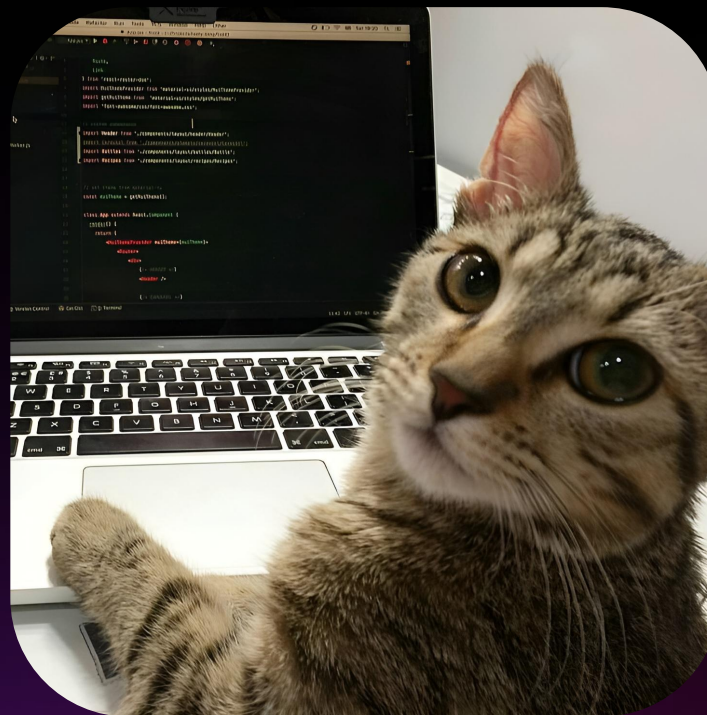
 Нефункциональные требования;

 Корнер-кейсы;

 Out of scope;

 Критерии приемки;

# Live coding



# Последствия

- ✓ Конкретный артефакт постановки задачи;

# Последствия

- ✓ Конкретный артефакт постановки задачи;
- ✓ Возможность провалидировать поставку задачи до реализации;

# Последствия

- ✓ Конкретный артефакт постановки задачи;
- ✓ Возможность провалидировать поставку задачи до реализации;
- ✓ Более предсказуемый результат ответа LLM;

# Последствия

✗ Тщательный анализ спеки требует времени;

# Последствия

- ✗ Тщательный анализ спеки требует времени;
- ✗ Важно следить за качеством описания спеки;

# Выводы



Claude Code - продвинутый ИИ-агент для автоматизации инженерной работы;

# Выводы



Claude Code - продвинутый ИИ-агент для автоматизации инженерной работы;



Качество результата определяется не только моделью, но и организацией контекста;

# Выводы



Claude Code - продвинутый ИИ-агент для автоматизации инженерной работы;



Качество результата определяется не только моделью, но и организацией контекста;



Skills, Subagents и CLAUDE.md позволяют улучшить качество работы с LLM;

# Выводы



Claude Code - продвинутый ИИ-агент для автоматизации инженерной работы;



Качество результата определяется не только моделью, но и организацией контекста;



Skills, Subagents и CLAUDE.md позволяют улучшить качество работы с LLM;



Работа с ИИ-агентами — это отдельный инженерный workflow;

# Ссылки на материалы

<https://github.com/glebkacode/StroitelApp>

<https://arxiv.org/abs/2509.21361>

<https://www.trychroma.com/research/context-rot>

<https://code.claude.com/docs/en/sub-agents>

<https://code.claude.com/docs/en/skills>

---

# Спасибо за внимание!



@JaJaGleb



GitHub

КИОН