Организация правил работы с проектами в Claude

Денис Савицкий, dev lead, Deltasoft

Специфика

- 10 параллельных проектов разной сложности
- Переключение разработчиков между проектами
- Большая и запутанная кодовая база
- Чувствительные к безопасности алгоритмы и данные
- Частое переключение контекста
- Почти все могут понимать английский + англоговорящие заказчики
- Частые задачи “Собрать MVP для …”
- Самостоятельные DevOps процессы

Основные инструменты разработки

- Оптимизируем потребление токенов (caveman/ponytail)
  - ENG для CLI
    - https://habr.com/ru/companies/gptunnel/articles/986526/ (rus/en 50%)
    - https://vc.ru/ai/2954286-russkiy-yazyk-v-ai-mify-o-stoimosti-tokenov-i-realnost (rus/en 25-30%)
  - CodeGraph - index кода, быстрый поиск и создание контекста задачи агентом
  - Prompt Rule: Task Role Context Expectation
  - CLAUDE.md - общий и проектные, отдельные сессии для работы
    - https://github.com/multica-ai/andrej-karpathy-skills
  - /usage для статистики

- SKILLS - plugin “skill-creator”, /insights
  - ansible-infra
  - codereview
  - git-split-commits
  - pr-description
  - rspec-test
  - server-triage
  - slack-deploy-notify
  - uptime-kuma

- gemini.google.com/app создаёт план разработки и декомпозиции задач по ТЗ
  - "create an implementation plan for claude to create a ruby on rails project completing simple implementation of amazon-like s3 storage, be aware of common production launch problems, scalability and clean architecture, split implementation in phases, use this technical specification and requirements: tech_task.md."

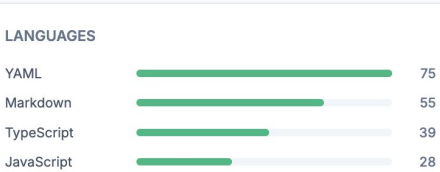
Claude session

1. claude 'complete task and give'
2. /rename api-proxy-backend
    a. claude --resume "api-proxy-backend"
3. plan-mode, больше контроля и безопасности
4. /init - создаёт CLAUDE.md со знаниями о проекте
5. /insights - создаёт страницу сводку о сессии
   подсказывает, что можно улучшить,
   подсвечивает проблемы,
   скелет выявленных SKILLS,
   промпты для исправления ошибок предыдущих сессий

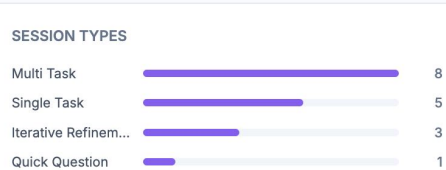
WHAT YOU WANTED

TOP TOOLS USED

LANGUAGES



SESSION TYPES



Custom Skills

Reusable /commands defined as markdown files.

Why for you: You already use a slack-deploy-notify skill successfully; codifying your repeated GitHub Actions Slack notification workflow and PR-writing into more skills would speed up your most common tasks.

```
# .claude/skills/pr/SKILL.md
Write a PR description and title.
- Title MUST use a conventional commit prefix.
- Include summary, changes by file, and test results.
- Verify the branch is up to date with main first.
```

Copy

MCP Servers

Connect Claude to external tools and APIs via Model Context Protocol.

Why for you: Your Uptime Kuma, Slack, and Telegram integration sessions involved manual API/socketio debugging; an MCP server for these APIs would let Claude query them directly instead of guessing.

```
claude mcp add github -- npx -y @modelcontextprotocol/server-github
```

Copy

Where Things Go Wrong

Your sessions mostly succeed, but friction clusters around premature execution before context is established, environment/tooling mismatches, and getting interrupted during exploratory phases.

Premature action before context is set

Claude sometimes starts implementing or exploring before the project context, branch, or scope is fully established, leading to rework or interruptions. You can reduce this by stating constraints (directory, branch, package manager, scope) up front and asking Claude to confirm a plan before touching files.

- Claude started exploring/initializing in the wrong directory and was interrupted twice; ultraplan then failed due to a missing git repo, and the session ended before code was written.
- Initial implementation began before you requested a branch recheck and DataQualityChecker addition, forcing a follow-up fix.

Environment and tooling mismatches

Several failures came from Claude using the wrong tool or missing project-specific conventions, breaking your environment. Specifying your toolchain and conventions (package manager, commit format, secrets handling) in CLAUDE.md or your initial prompt would prevent these cleanup detours.

- Claude ran npm install instead of pnpm, causing a dependency conflict that broke the dev server and required cleanup before verification.
- An initial PR title lacked a conventional commits prefix, and a workflow refactor used secrets.* in an if: condition plus mishandled multiline commit messages — each requiring you to point out the errors.

Abandoned exploratory sessions and transient errors

Multiple sessions ended with nothing produced because you interrupted long exploration or planning phases, and transient API errors occasionally forced resubmissions. Letting Claude commit to a concrete first action faster — or scoping tasks more narrowly — would help these sessions reach completion.

- You interrupted Claude's Agent exploration and AskUserQuestion calls and exited before CLAUDE.md or deployment prep was ever produced (not_achieved outcomes).
- Two transient API connection/auth errors on the OpenWrt request forced you to resubmit the same message three times.

Security

Команды для CLI чтобы посмотреть доступы:

- /permissions
- /allowed-tools

Записать все разрешения и запреты:

~/.claude/settings.json
<project>/.claude/settings.json

```
{
  "permissions": {
    "allow": [
      "Bash(npm run lint)",
      "Bash(npm run test:*)",
      "Read(~/.zshrc)"
    ],
    "deny": [
      "Read(**/.env*)",
      "Read(**/*.pem)",
      "Read(**/*.key)",
      "Read(**/secrets/**)",
      "Read(**/credentials/**)",
      "Read(**/.aws/**)",
      "Read(**/.ssh/**)",
      "Read(**/docker-compose*.yml)",
      "Read(**/config/database.yml)"
    ]
  }
}
```

Plugins

- https://github.com/colbymchenry/codegraph
  index, помогает агенту быстрее находить кратчайший путь до классов, функций и структуры кода

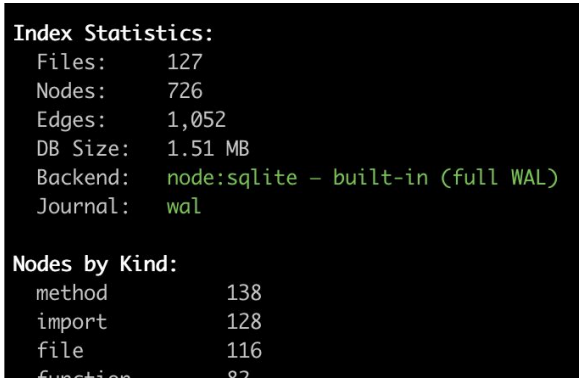
  codegraph init i; codegraph serve --mcp

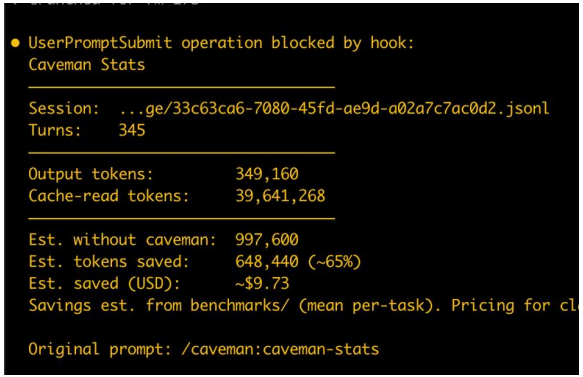
- https://github.com/JuliusBrussee/caveman
  CLI общается в стиле пещерного человека, экономит больше 50% токенов
  альтернатива: https://github.com/DietrichGebert/ponytail

- https://claude.com/plugins/security-guidance
  Официальный плагин Anthropic, подсвечивает XSS, небезопасные паттерны в коде

- https://github.com/Digital-Process-Tools/claude-remember

- https://github.com/Egonex-AI/Understand-Anything
  не рекомендую, на простом проекте ничего полезного не показал, сожрав почти 100k токенов





Полезно посмотреть

https://www.youtube.com/watch?v=-IozMG9x0dI

/insights — анализирует сессию и смотрит, какие были допущены ошибки, какие SKILLS можно добавить для регулярных задач, как лучше формировать промпты

https://www.youtube.com/watch?v=ihVShDkmQyM

Создавать CONTEXT.md внутри проекта

/discuss idea — использует контекст для вопросов по проекту

Вопросы?

@sadfuzzy

github.com/**sadfuzzy**