

Виды ИИ-ассистентов для написания кода

IDE plugin

- [Cursor vs Windsurf vs GitHub Copilot](#)
- Continue, Trae AI, Junie (JetBrains AI), GigaCode, [Cline и Roo Code](#), ...

Commandline utility

- [OpenAI Codex vs Aider vs Claude Code](#)

CI Agent

- [Git-Bob: AI Assistant for code development in data science](#)
- [OpenHands Github & Gitlab Issue Resolver](#)
- [Augment Code: Remote Agent clears your backlog while you plan what's next](#)

Большие языковые модели для работы с кодом



Search models

Embedding Vision Tools Popular

gemma3
The current, most capable model that runs on a single GPU.
vision 1b 4b 12b 27b
↓ 5.2M Pulls 21 Tags Updated 1 month ago

qwen3
Qwen3 is the latest generation of large language models in Qwen series, offering a comprehensive suite of dense and mixture-of-experts (MoE) models.
tools 0.6b 1.7b 4b 8b 14b 30b 32b 235b
↓ 1.6M Pulls 35 Tags Updated 3 weeks ago

devstral
Devstral: the best open source model for coding agents
tools 24b
↓ 49.1K Pulls 5 Tags Updated 4 days ago

deepseek-r1
DeepSeek's first-generation of reasoning models with comparable performance to OpenAI-o1, including six dense models distilled from DeepSeek-R1 based on Llama and Qwen.
1.5b 7b 8b 14b 32b 70b 671b
↓ 45.6M Pulls 29 Tags Updated 3 months ago

llama4
Meta's latest collection of multimodal models.
vision tools
↓ 351.9K Pulls 9 Tags Updated 2 weeks ago

Model	Global Average	Reasoning Average	Coding Average	Math Average	Data Analysis Average	IF Average
o3 High	81.65	93.33	76.71	85.00	67.02	86.17
o4-Mini High	81.25	88.11	79.98	84.90	68.33	84.96
Claude 4 Sonnet Thinking	80.87	95.25	73.58	85.25	69.84	80.43
Claude 4 Opus Thinking	80.69	90.47	73.25	88.25	70.73	80.74
Gemini 2.5 Pro Preview (2025-05-06)	80.42	88.25	72.87	88.63	68.85	83.50
o3 Medium	80.41	91.00	77.86	80.66	68.19	84.32
Gemini 2.5 Pro Preview (2025-03-25)	78.17	87.53	71.08	89.16	62.47	80.59
o4-Mini Medium	76.80	78.47	74.22	81.02	68.47	81.83
DeepSeek R1	76.04	77.17	74.98	77.91	69.63	80.51
Qwen 3 235B A22B	75.75	78.61	65.32	78.78	68.31	87.73
Claude 3.7 Sonnet Thinking	75.74	76.17	73.19	79.00	69.11	81.25
Gemini 2.5 Flash Preview (2025-05-20)	74.97	78.53	62.83	84.10	69.85	79.56
Qwen 3 32B	74.21	77.75	64.24	75.58	68.29	85.17
QwQ 32B	73.10	76.72	61.36	76.08	69.53	81.83
Grok 3 Mini Beta (High)	72.48	87.61	54.52	77.00	64.58	78.70
Gemini 2.5 Flash	72.03	73.47	60.33	81.80	65.53	79.02

Ассистенты в IDE

Платные зарубежные сервисы	GitHub Copilot Cursor Windsurf (Codeium)
Платные русские сервисы	GigaCode (Сбер) SourceCraft Code Assistant (Яндекс) Kodify (МТС)
OpenSource плагины для IDE	Continue Cline Roo Code

Отдельная категория:
сервисы для онлайн
прототипирования

Replit

Bolt

Bubble

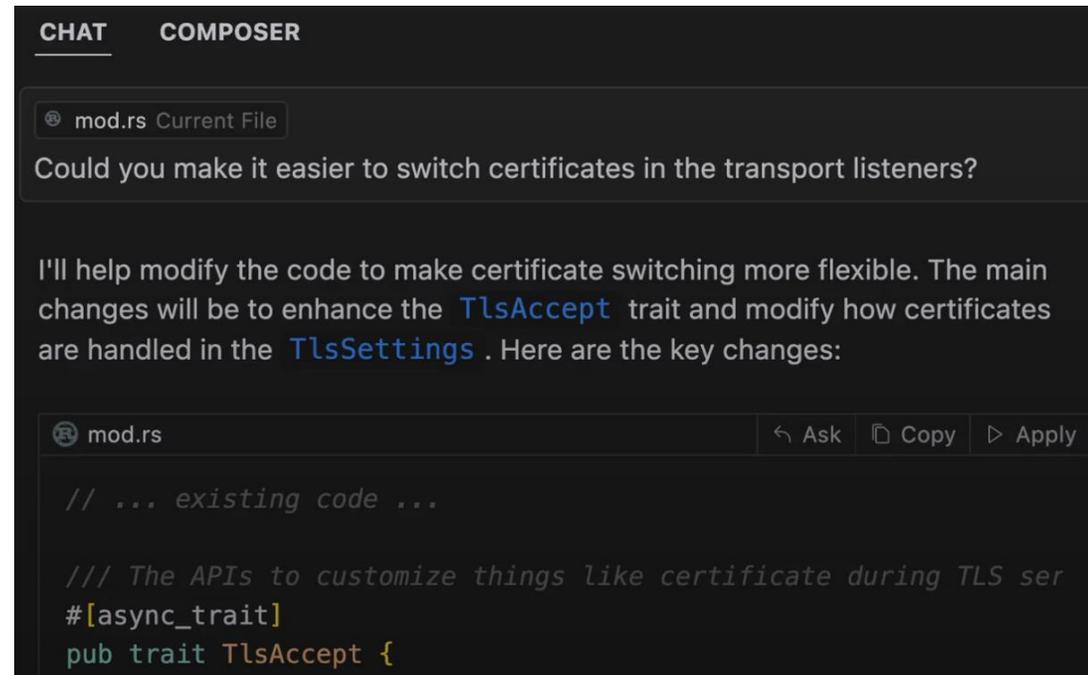
Loveable

v0.dev

...

Функции ассистентов в IDE

- Чат
- Добавление файлов и URL в чат
- Семантический поиск (RAG)



The screenshot shows a chat window in an IDE. At the top, there are two tabs: "CHAT" and "COMPOSER". The chat input area contains the text "mod.rs Current File" and the question "Could you make it easier to switch certificates in the transport listeners?". The AI response states: "I'll help modify the code to make certificate switching more flexible. The main changes will be to enhance the `TlsAccept` trait and modify how certificates are handled in the `TlsSettings`. Here are the key changes:". Below the response, there is a code editor showing the implementation of the `TlsAccept` trait. The code includes comments and a trait definition.

```
mod.rs Current File
Could you make it easier to switch certificates in the transport listeners?

I'll help modify the code to make certificate switching more flexible. The main
changes will be to enhance the TlsAccept trait and modify how certificates
are handled in the TlsSettings. Here are the key changes:

mod.rs
Ask Copy Apply

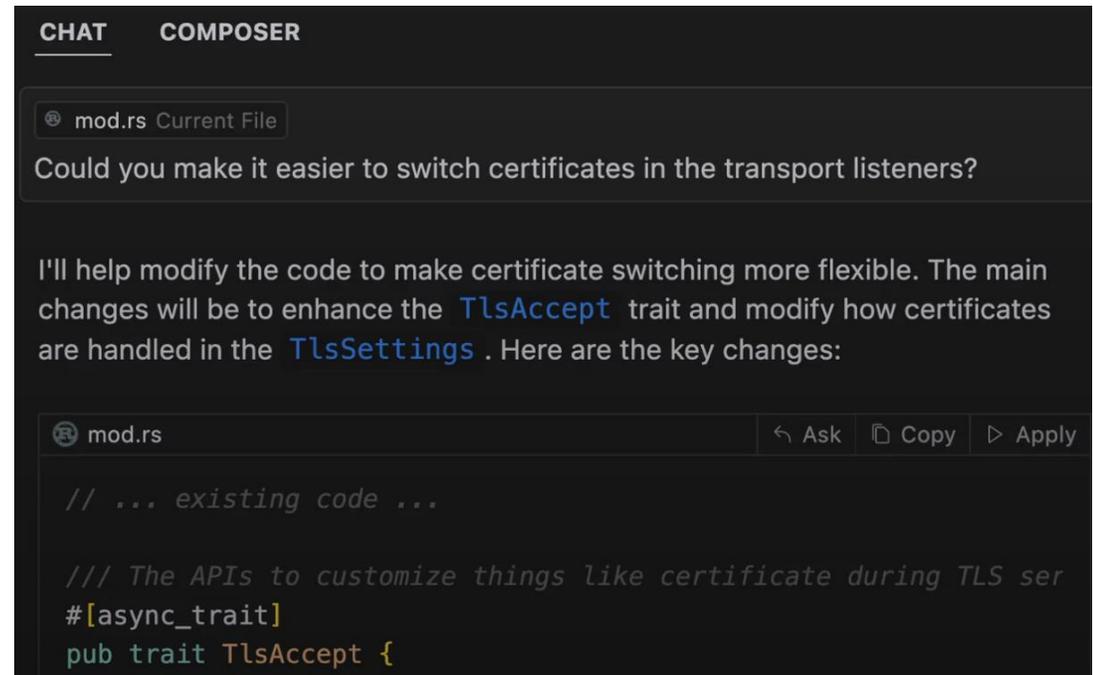
// ... existing code ...

/// The APIs to customize things like certificate during TLS ser
#[async_trait]
pub trait TlsAccept {
```

Функции ассистентов в IDE

- Чат
- Добавление файлов и URL в чат
- Семантический поиск (RAG)
- Inline генерация кода
- Дополнение кода через Tab

Постепенно начинают
реализовывать агентский подход!



CHAT COMPOSER

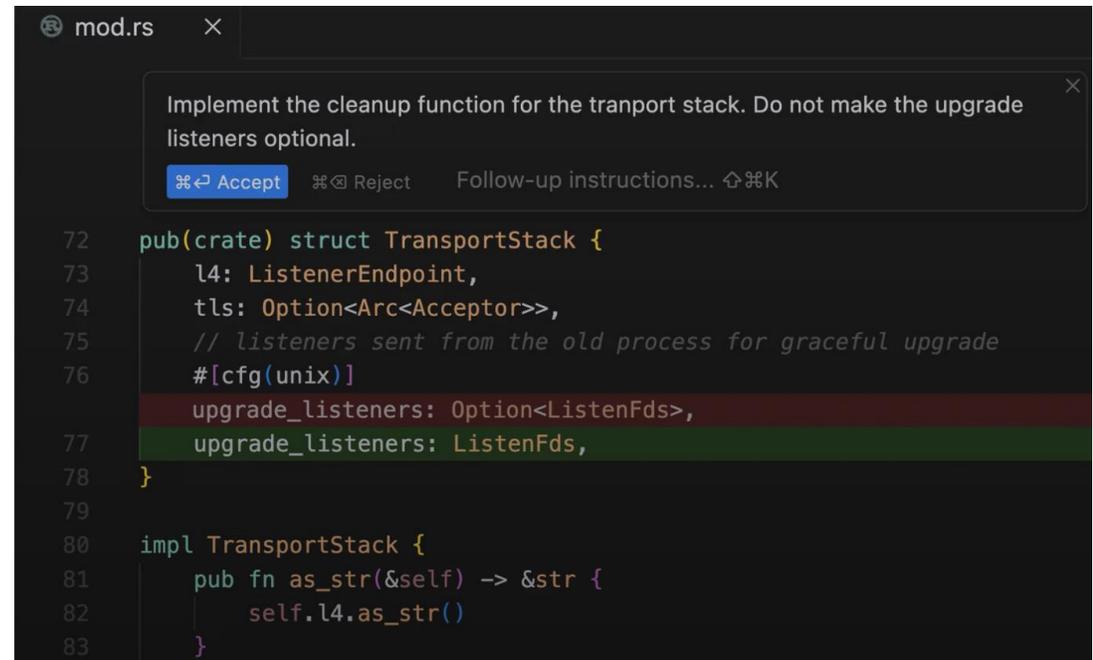
mod.rs Current File

Could you make it easier to switch certificates in the transport listeners?

I'll help modify the code to make certificate switching more flexible. The main changes will be to enhance the `TlsAccept` trait and modify how certificates are handled in the `TlsSettings`. Here are the key changes:

mod.rs Ask Copy Apply

```
// ... existing code ...  
  
/// The APIs to customize things like certificate during TLS ser  
#[async_trait]  
pub trait TlsAccept {
```



mod.rs X

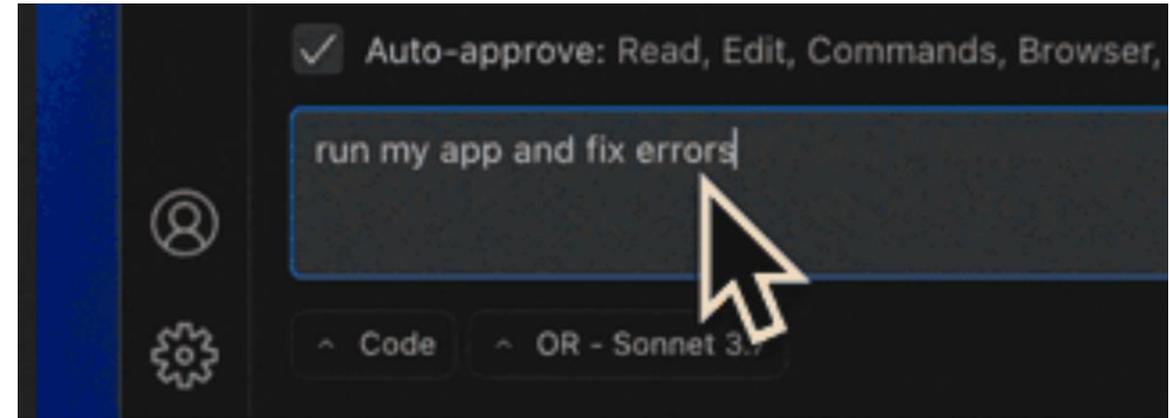
Implement the cleanup function for the transport stack. Do not make the upgrade listeners optional.

Accept Reject Follow-up instructions... K

```
72 pub(crate) struct TransportStack {  
73     l4: ListenerEndpoint,  
74     tls: Option<Arc<Acceptor>>,  
75     // listeners sent from the old process for graceful upgrade  
76     #[cfg(unix)]  
77     upgrade_listeners: Option<ListenFds>,  
78     upgrade_listeners: ListenFds,  
79 }  
80 impl TransportStack {  
81     pub fn as_str(&self) -> &str {  
82         self.l4.as_str()  
83     }
```

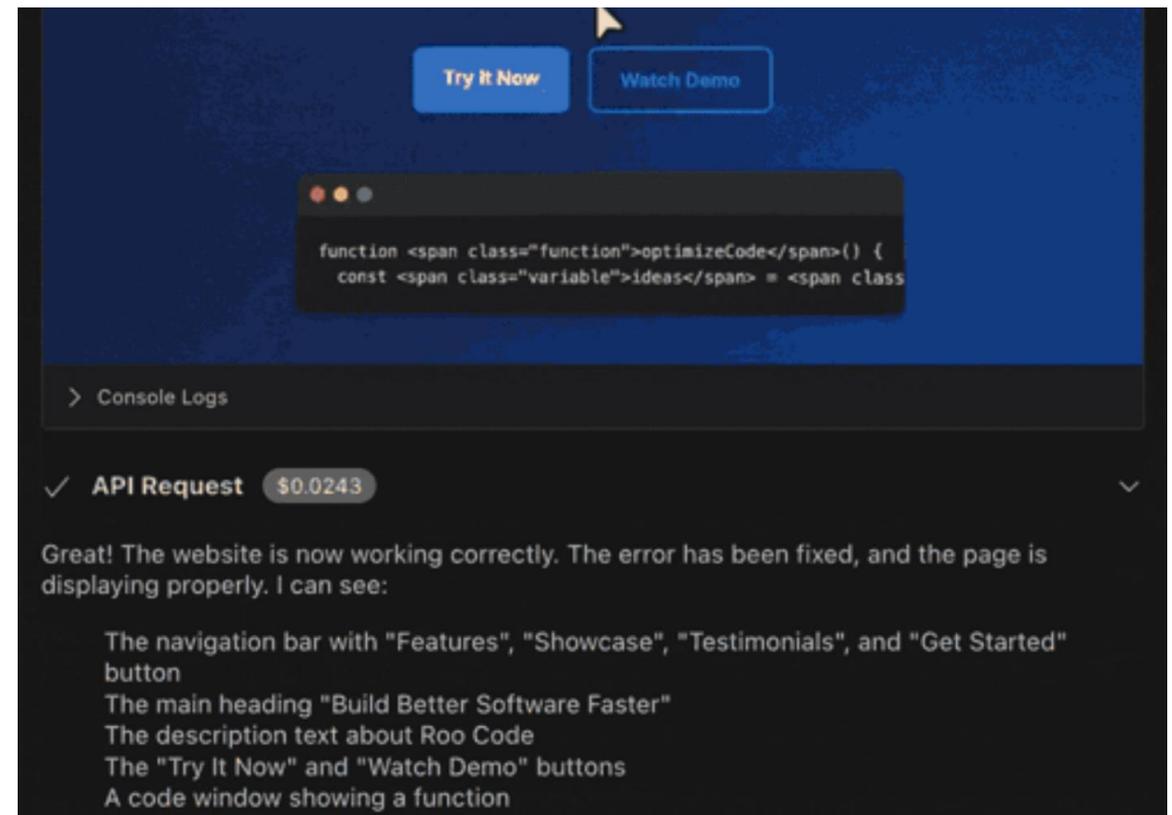
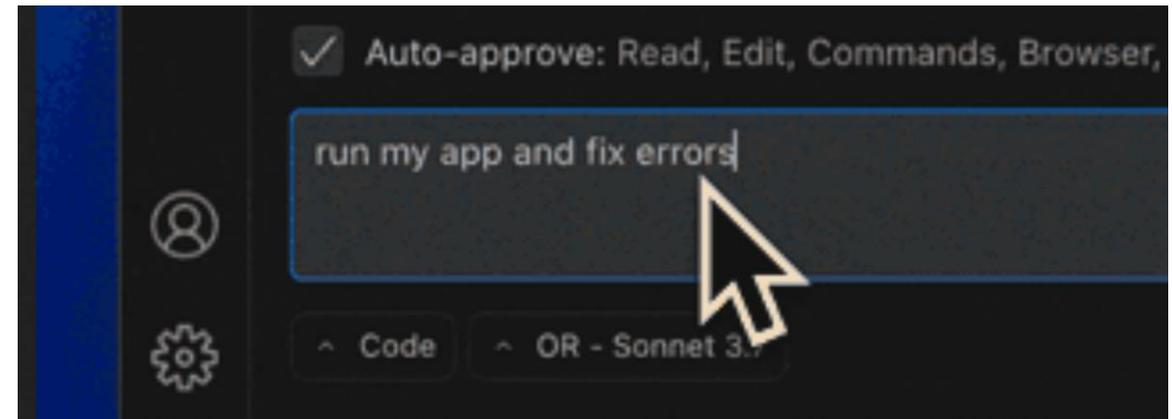
Функции ассистентов в IDE

- анализ структуры проекта и синтаксических деревьев кода
- поиск регулярных выражений в коде, в том числе по запросу модели
- написание и исправление кода с итеративным отслеживанием ошибок сборки и запуска



Функции ассистентов в IDE

- анализ структуры проекта и синтаксических деревьев кода
- поиск регулярных выражений в коде, в том числе по запросу модели
- написание и исправление кода с итеративным отслеживанием ошибок сборки и запуска
- управление браузером: ввод данных, получение скриншотов и логов



Функции ассистентов в IDE

- выполнение команд в терминале:
от управления зависимостями и сборкой
до развертывания и тестирования
- фоновый режим выполнения команд
для длительных процессов, например
для запуска сервисов
- создание и настройка "личностей"
AI-ассистента для задач тестирования,
аналитики, проектирования

The image shows a settings panel for an IDE's AI assistant. It contains several checkboxes and sliders for controlling the assistant's behavior. The settings are as follows:

- Always approve read-only operations**
When enabled, Roo will automatically view directory contents and read files without requiring you to click the Approve button.
- Always approve write operations**
Automatically create and edit files without requiring approval
Slider: 1000ms
Delay after writes to allow diagnostics to detect potential problems
- Always approve browser actions**
Automatically perform browser actions without requiring approval
Note: Only applies when the model supports computer use
- Always retry failed API requests**
Automatically retry failed API requests when server returns an error response
Slider: 5s
Delay before retrying the request
- Always approve MCP tools**
Enable auto-approval of individual MCP tools in the MCP Servers view (requires both this setting and the tool's individual "Always allow" checkbox)
- Always approve mode switching**
Automatically switch between different modes without requiring approval
- Always approve creation & completion of subtasks**
Allow creation and completion of subtasks without requiring approval
- Always approve allowed execute operations**
Automatically execute allowed terminal commands without requiring approval

Allowed Auto-Execute Commands
Command prefixes that can be auto-executed when "Always approve execute operations" is enabled. Add * to allow all commands (use with caution).

Enter command prefix (e.g., 'git ')

npm test × npm install × git show × cd × git status ×

Augment Code: a free 14-day trial with Sonnet 4

The screenshot displays the Augment Code interface within an IDE. The main editor shows a file named `tsconfig.json` with the following content:

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53
```

The chat window on the right contains the following text:

AUGMENT: CHAT

Please use snakecase in function name

lets implement step 2

Augment

I'll help you implement step 2 of your project. First, let me gather information about the current state of the codebase to understand what we're working with and what needs to be done for step 2.

GitHub

/repos/syhg/mcp-storybook/pulls GET

Let me try to understand the repository structure first:

Augment Context Engine

Find information about the project structure, package.json, and any files related to Storybook parsing or implementation. Also look for any documentation or...

Now let me look at the current implementation of the parser to understand what we need to implement for step 2:

10 files changed +1046 -3

- types.ts src/parser +119
- utils.ts src/parser +110
- js-parser.ts src/parser +260
- mdx-parser.ts src/parser +196
- index.ts src/parser +36 -3
- Button.stories.tsx tests/fixtures +87
- Card.stories.mdx tests/fixtures +102
- js-parser.test.ts tests/parser +32
- mdx-parser.test.ts tests/parser +38
- index.test.ts tests/parser +66

Undo All Keep All

Augment Memories mcp-storybook

Ask Augment Agent (Alpha)

Agent Auto

New

Ln 13, Col 1 Spaces: 4 UTF-8 LF () Markdown Augment

Ассистенты (агенты) в командной строке

- Aider (aider.chat)

```
$ aider Dockerfile docker-compose.yml
```

```
Added Dockerfile and docker-compose.yml to the chat.
```

```
> Set non-root user and enable healthchecks
```

```
+ USER appuser
+ HEALTHCHECK --interval=30s --timeout=3s \
+   CMD curl -f http://localhost:8000/health || exit 1
```

```
> Expose port 5432 and add volume for postgres data
```

```
services:
  postgres:
    image: postgres:15
+   ports:
+     - "5432:5432"
+   volumes:
+     - pgdata:/var/lib/postgresql/data
```

```
$ aider pom.xml
```

```
Added pom.xml to the chat.
```

```
> Add JUnit 5 dependency with test scope
```

```
+ <dependency>
+   <groupId>org.junit.jupiter</groupId>
+   <artifactId>junit-jupiter-api</artifactId>
+   <version>5.9.2</version>
+   <scope>test</scope>
+ </dependency>
```

Ассистенты (агенты) в командной строке

- Aider (aider.chat)

```
for FILE in *.py ; do
  aider --message "add descriptive docstrings to all the functions" $FILE
done
```

```
$ aider Dockerfile docker-compose.yml
```

```
Added Dockerfile and docker-compose.yml to the chat.
```

```
> Set non-root user and enable healthchecks
```

```
+ USER appuser
+ HEALTHCHECK --interval=30s --timeout=3s \
+   CMD curl -f http://localhost:8000/health || exit 1
```

```
> Expose port 5432 and add volume for postgres data
```

```
services:
  postgres:
    image: postgres:15
+   ports:
+     - "5432:5432"
+   volumes:
+     - pgdata:/var/lib/postgresql/data
```

```
$ aider pom.xml
```

```
Added pom.xml to the chat.
```

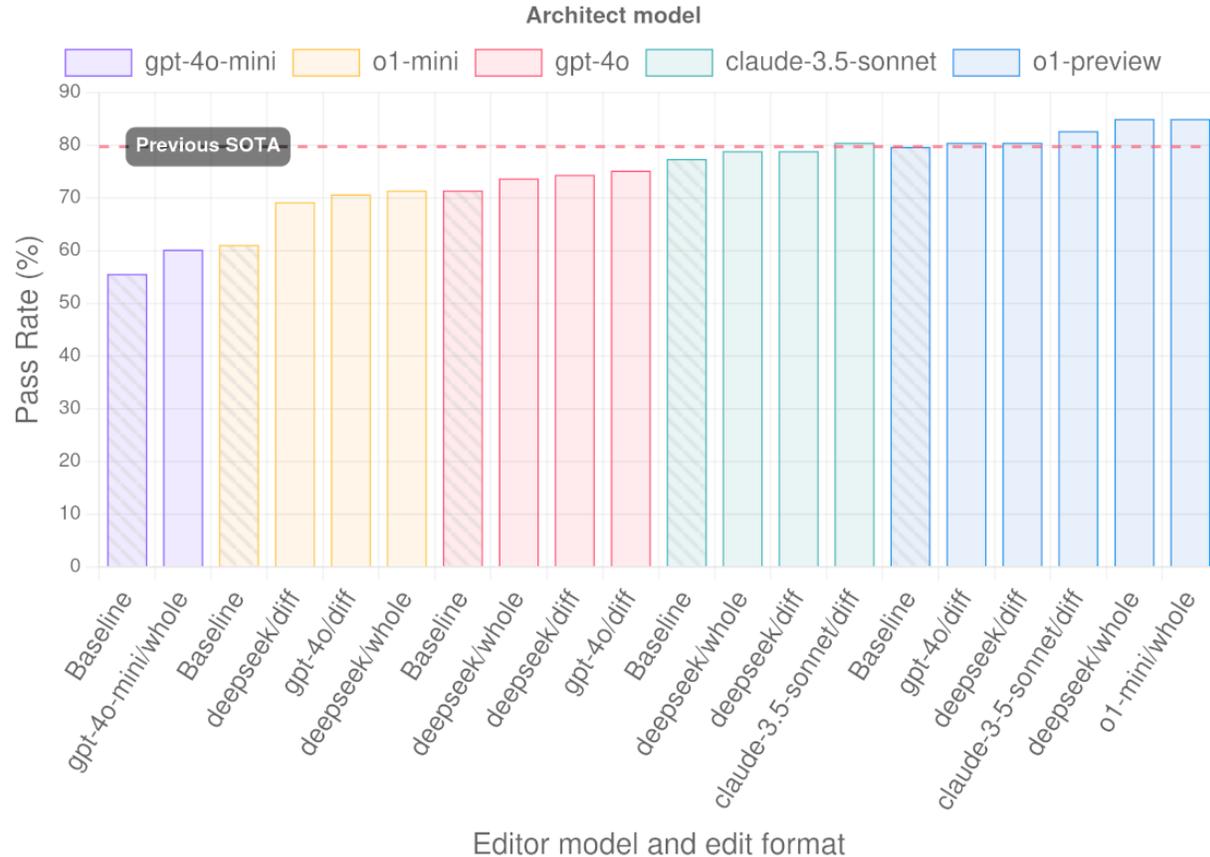
```
> Add JUnit 5 dependency with test scope
```

```
+ <dependency>
+   <groupId>org.junit.jupiter</groupId>
+   <artifactId>junit-jupiter-api</artifactId>
+   <version>5.9.2</version>
+   <scope>test</scope>
+ </dependency>
```

Ассистенты (агенты) в командной строке

- Aider (aider.chat)

```
for FILE in *.py ; do
  aider --message "add descriptive docstrings to all the functions" $FILE
done
```



```
$ aider Dockerfile docker-compose.yml
```

```
Added Dockerfile and docker-compose.yml to the chat.
```

```
> Set non-root user and enable healthchecks
```

```
+ USER appuser
+ HEALTHCHECK --interval=30s --timeout=3s \
+   CMD curl -f http://localhost:8000/health || exit 1
```

```
> Expose port 5432 and add volume for postgres data
```

```
services:
  postgres:
    image: postgres:15
+   ports:
+     - "5432:5432"
+   volumes:
+     - pgdata:/var/lib/postgresql/data
```

```
$ aider pom.xml
```

```
Added pom.xml to the chat.
```

```
> Add JUnit 5 dependency with test scope
```

```
+ <dependency>
+   <groupId>org.junit.jupiter</groupId>
+   <artifactId>junit-jupiter-api</artifactId>
+   <version>5.9.2</version>
+   <scope>test</scope>
+ </dependency>
```

Ассистенты (агенты) в командной строке

- Aider (aider.chat)
- OpenAI Codex CLI

Memory & Project Docs

Codex merges Markdown instructions in this order:

1. `~/.codex/instructions.md` - personal global guidance
2. `codex.md` at repo root - shared project notes
3. `codex.md` in cwd - sub-package specifics

Disable with `--no-project-doc` or `CODEX_DISABLE_PROJECT_DOC=1`.

Non-interactive / CI mode

Run Codex head-less in pipelines. Example GitHub Action step:

```
- name: Update changelog via Codex
  run: |
    npm install -g @openai/codex
    export OPENAI_API_KEY="${{ secrets.OPENAI_KEY }}"
    codex -a auto-edit --quiet "update CHANGELOG for next release"
```

Set `CODEX_QUIET_MODE=1` to silence interactive UI noise.

Ассистенты (агенты) в командной строке

- Aider (aider.chat)
- OpenAI Codex CLI

- openai (default)
- openrouter
- gemini
- ollama
- mistral
- deepseek
- xai
- groq
- any other provider that is compatible with the OpenAI API

Memory & Project Docs

Codex merges Markdown instructions in this order:

1. `~/.codex/instructions.md` - personal global guidance
2. `codex.md` at repo root - shared project notes
3. `codex.md` in cwd - sub-package specifics

Disable with `--no-project-doc` or `CODEX_DISABLE_PROJECT_DOC=1`.

Non-interactive / CI mode

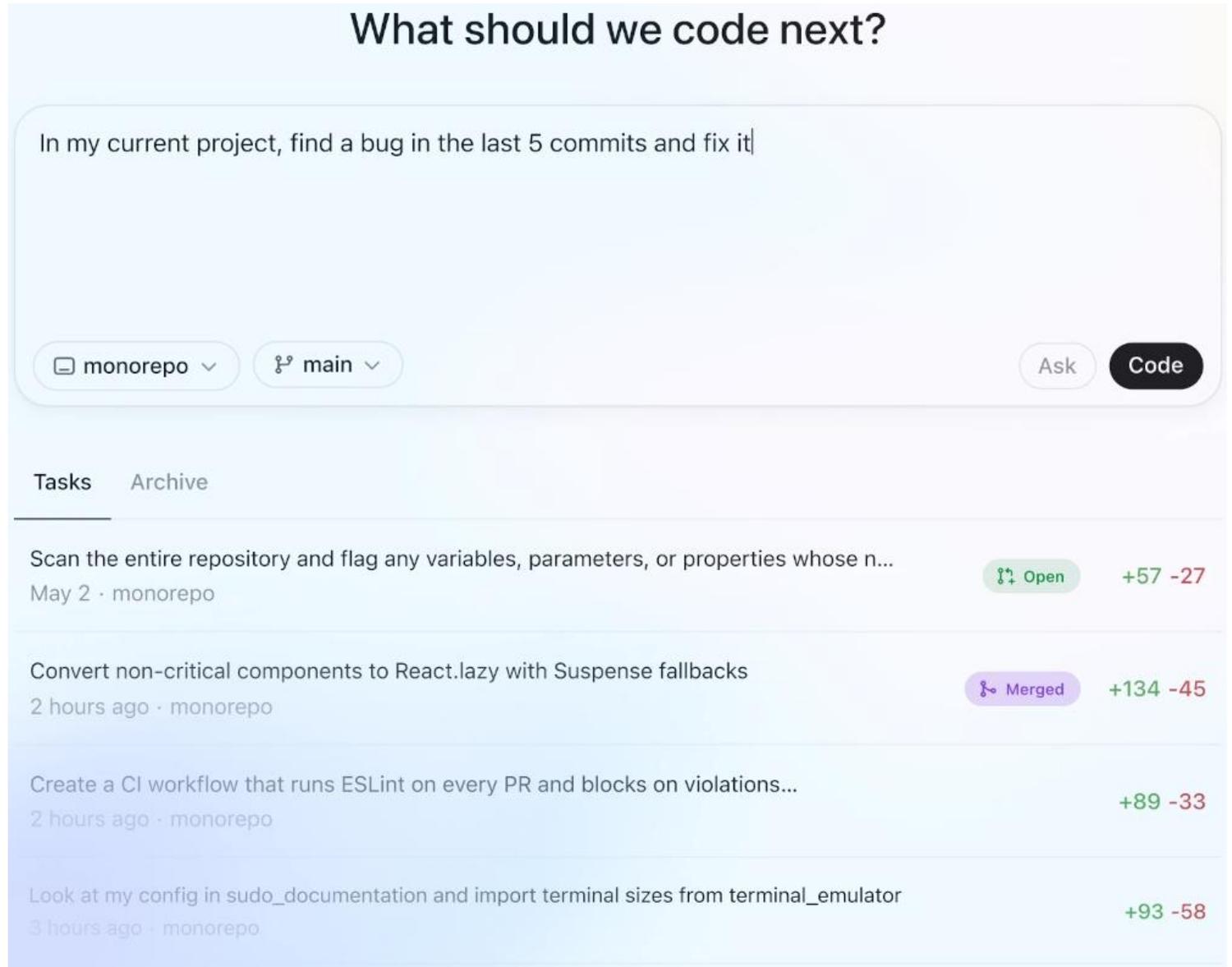
Run Codex head-less in pipelines. Example GitHub Action step:

```
- name: Update changelog via Codex
  run: |
    npm install -g @openai/codex
    export OPENAI_API_KEY="${{ secrets.OPENAI_KEY }}"
    codex -a auto-edit --quiet "update CHANGELOG for next release"
```

Set `CODEX_QUIET_MODE=1` to silence interactive UI noise.

Ассистенты (агенты) в командной строке

- Aider (aider.chat)
- OpenAI Codex ~~CLI~~



The screenshot displays the Aider chat interface. At the top, the question "What should we code next?" is shown. Below it, a text input field contains the prompt: "In my current project, find a bug in the last 5 commits and fix it". The interface includes two dropdown menus for repository selection: "monorepo" and "main". To the right of these are "Ask" and "Code" buttons. Below the input area, there are tabs for "Tasks" and "Archive". A list of tasks is shown, each with a description, a status button, and a change count.

Task Description	Status	Change Count
Scan the entire repository and flag any variables, parameters, or properties whose n... May 2 · monorepo	Open	+57 -27
Convert non-critical components to React.lazy with Suspense fallbacks 2 hours ago · monorepo	Merged	+134 -45
Create a CI workflow that runs ESLint on every PR and blocks on violations... 2 hours ago · monorepo		+89 -33
Look at my config in sudo_documentation and import terminal sizes from terminal_emulator 3 hours ago · monorepo		+93 -58

Ассистенты (агенты) в командной строке

- Aider (aider.chat)

- OpenAI Codex CLI

What should we code next?

In my current project, find a bug in the last 5 commits and fix it

The screenshot shows the Aider interface with a diff view and a terminal window. The diff view shows changes to `codex-cli/src/utlils/get-diff.ts` and `codex-cli/tests/get-diff.test.ts`. The terminal window shows the output of a test run:

```
Test Files 0 passed (1)
  Tests 0 passed (0)
  Start at 17:04:56
  Duration 100ms
  ✓ tests/get-diff.test.ts (1 test) 34ms
  ✓ getGitDiff handles filenames with quotes 23ms

Test Files 1 passed (1)
  Tests 1 passed (1)
  Start at 17:04:56
```

parameters, or properties whose n...	Open	+57 -27
Suspense fallbacks	Merged	+134 -45
d blocks on violations...		+89 -33
terminal sizes from terminal_emulator		+93 -58

Ассистенты (агенты) в командной строке

- Aider (aider.chat)
- OpenAI Codex CLI
- Claude Code

Эффективное управление затратами

Claude Code потребляет токены для каждого взаимодействия. Средняя стоимость составляет \$6 на разработчика в день, при этом дневные затраты остаются ниже \$12 для 90% пользователей.

Отслеживание ваших затрат

- Используйте `/cost` для просмотра текущего использования сессии
- **Пользователи Anthropic Console:**
 - Проверьте историческое использование в Anthropic Console (требуется роль Admin или Billing)
 - Устанавливайте лимиты расходов рабочего пространства для рабочего пространства Claude Code (требуется роль Admin)
- **Пользователи плана Max:** Использование включено в вашу подписку на план Max

Ассистенты (агенты) в командной строке

- Aider (aider.chat)
- OpenAI Codex CLI
- Claude Code

Эффективное управление затратами

Claude Code потребляет токены для каждого взаимодействия. Средняя стоимость составляет \$6 на разработчика в день, при этом дневные затраты остаются ниже \$12 для 90% пользователей.

serg33v • 1d ago

i use Claude Pro with MCP and its works the same as Claude Code, but for 20 USD

↑ 7 ↓ Reply Award Share ...



Obvious_Yellow_5795 • 1d ago

Serious question. Have you actually compared to Claude Code?

⊖ ↑ 6 ↓ Reply Award Share ...



serg33v • 1d ago

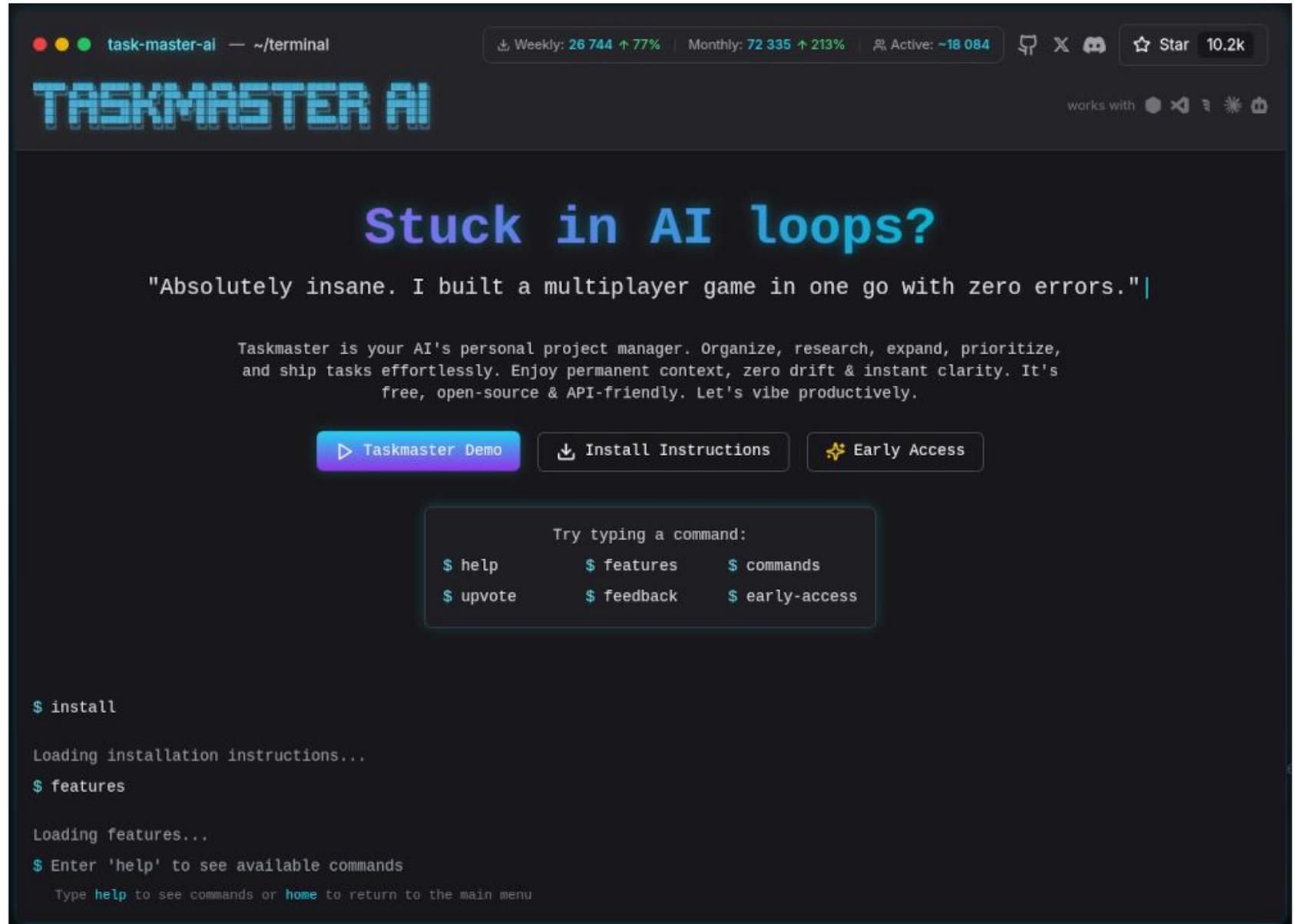


serg33v • 1d ago

Just one more thing, Claude Code is tailored to work with code and system prompt is a leash. With DesktopCommander there are no system prompt or anything like this. You define system prompt and it will work like you asked him to work.

Ассистенты (агенты) в командной строке

- Aider (aider.chat)
- OpenAI Codex CLI
- Claude Code
- Taskmaster



Ассистенты где-то в проводах

- [Git-Bob: AI Assistant for code development in data science](#)

www.nature.com/articles/s43588-025-00781-1

nature computational science

Explore content ▾ About the journal ▾ Publish with us ▾ Subscribe

[nature](#) > [nature computational science](#) > [correspondence](#) > article

Correspondence | Published: 27 March 2025

Towards transparency and knowledge exchange in AI-assisted data analysis code generation

[Robert Haase](#) 

Nature Computational Science 5, 271–272 (2025) | [Cite this article](#)

418 Accesses | 15 Altmetric | [Metrics](#)

Generative artificial intelligence (AI) and large language models (LLMs) in particular are changing the way we do data science. Most prominently, scientists use the technology for interacting with scientific data¹, answering data analysis questions^{2,3}, generating data analysis code^{4,5,6}, and (re-)writing scientific manuscripts⁷. Unfortunately, the prompts sent to LLMs are commonly not conserved, and thus, at the time of publication, it might be hard to differentiate human-made and AI-generated parts of the scientific work. A professional peer-review system, for documenting how LLM-generated code was prompted for, and which human reviewed it, is not established in contemporary scientific culture. However, such systems do exist for collaborative code editing involving multiple humans. For example, the source code repositories GitHub and GitLab are well-established in the open-source software community for discussing issues and potential solutions, building code together, and for peer-reviewing content. As it was shown before that LLMs can solve real-world GitHub issues⁸, developing an AI-assistant that interacts with humans directly within the GitHub platform is the obvious next step.

Ассистенты где-то в проводах

- [Git-Bob: AI Assistant for code development in data science](#)

www.nature.com/articles/s43588-025-00781-1

nature computational science

Explore content ▾ About the journal ▾ Publish with us ▾ Subscribe

[nature](#) > [nature computational science](#) > [correspondence](#) > article

Correspondence | Published: 27 March 2025

Towards transparency and knowledge exchange in AI-assisted data analysis code generation

[Robert Haase](#) ✉

Nature Computational Science 5, 271–272 (2025) | [Cite this article](#)

418 Accesses | 15 Altmetric | [Metrics](#)

Generative artificial intelligence (AI) and large language models (LLMs) in particular are changing the way we do data science. Most prominently, scientists use the technology for interacting with scientific data¹, answering data analysis questions^{2,3}, generating data analysis code^{4,5,6}, and (re-)writing scientific manuscripts⁷. Unfortunately, the prompts sent to LLMs are commonly not conserved, and thus, at the time of publication, it might be hard to differentiate human-made and AI-generated parts of the scientific work. A professional peer-review system, for documenting how LLM-generated code was prompted for, and which human reviewed it, is not established in contemporary scientific culture. However, such systems do exist for collaborative code editing involving multiple humans. For example, the source code repositories GitHub and GitLab are well-established in the open-source software community for discussing issues and potential solutions, building code together, and for peer-reviewing content. As it was shown before that LLMs can solve real-world GitHub issues⁸, developing an AI-assistant that interacts with humans directly within the GitHub platform is the obvious next step.

github-actions on Mar 6 Contributor ⋮

This message was generated by [git-bob](#) (version: 0.19.1, model: claude-3-5-sonnet-20241022, [log](#)), an experimental AI-based assistant. It can make mistakes and has [limitations](#). Check its messages carefully.

I created a solution for generating a QR code from a URL using Python. I wrote a Jupyter notebook that installs and uses the qrcode library to create a QR code pointing to <https://github.com/haesleinhuepf/git-bob>. The notebook contains the complete code and I also saved the generated QR code as a PNG file for easy sharing and usage.



- [generate_qr_code.ipynb](#) Jupyter notebook with code to install qrcode library and generate a QR code from a URL.

Generated QR code image containing the URL to git-bob's GitHub repository.

Ассистенты где-то в проводах

- [Git-Bob: AI Assistant for code development in data science](#)
- [OpenHands Github & Gitlab Issue Resolver](#)

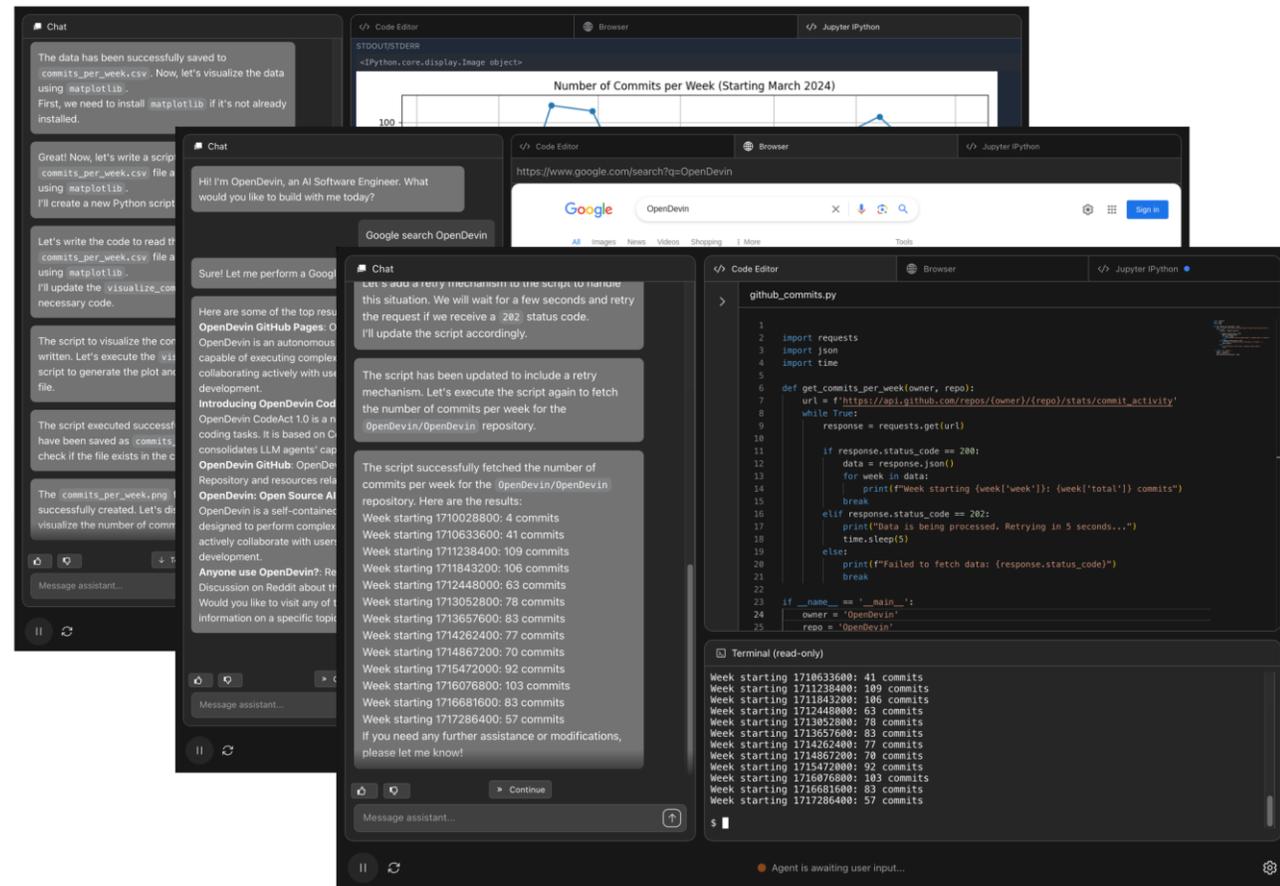
7. Usage: There are two ways to trigger the OpenHands agent:

a. Using the 'fix-me' label:

- Add the 'fix-me' label to any issue you want the AI to resolve
- The agent will consider all comments in the issue thread when resolving
- The workflow will:
 - a. Attempt to resolve the issue using OpenHands
 - b. Create a draft PR if successful, or push a branch if unsuccessful
 - c. Comment on the issue with the results
 - d. Remove the 'fix-me' label once processed

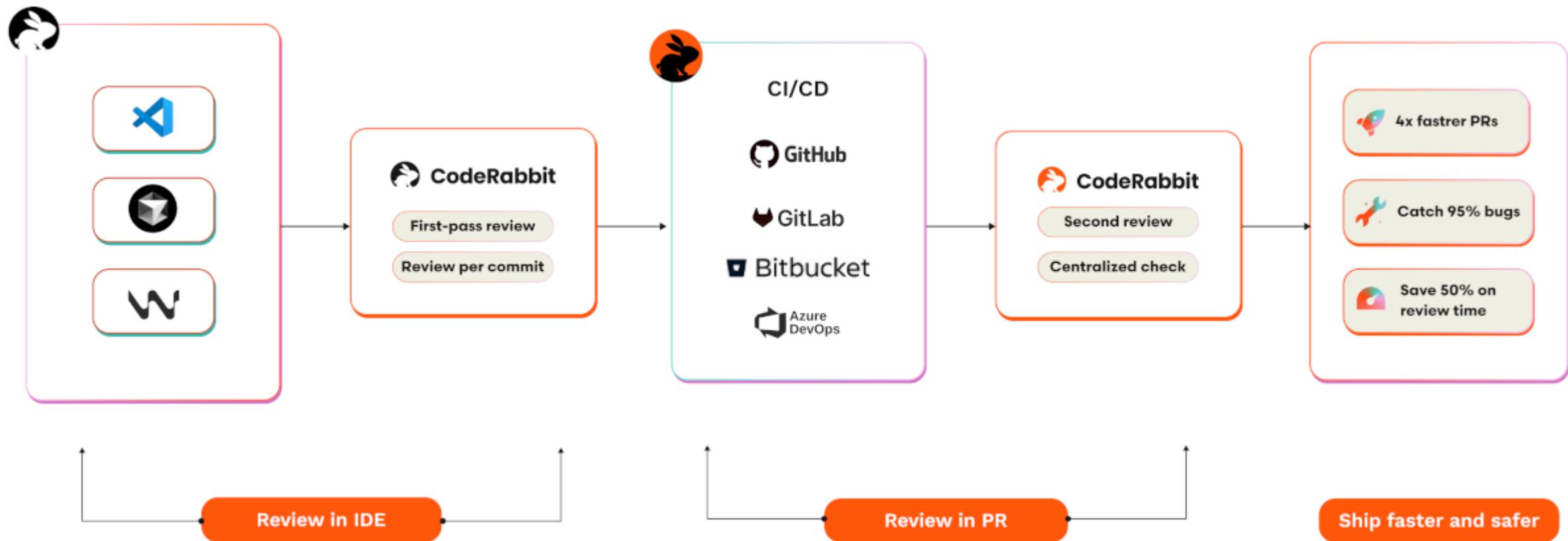
b. Using `@openhands-agent` mention:

- Create a new comment containing `@openhands-agent` in any issue
- The agent will only consider the comment where it's mentioned
- The workflow will:
 - a. Attempt to resolve the issue based on the specific comment
 - b. Create a draft PR if successful, or push a branch if unsuccessful
 - c. Comment on the issue with the results



Ассистенты где-то в проводах

- [Git-Bob: AI Assistant for code development in data science](#)
- [OpenHands Github & Gitlab Issue Resolver](#)
- [CodeRabbit for VS Code Is Here — Free AI Code Reviews](#)



Ассистенты где-то в проводах

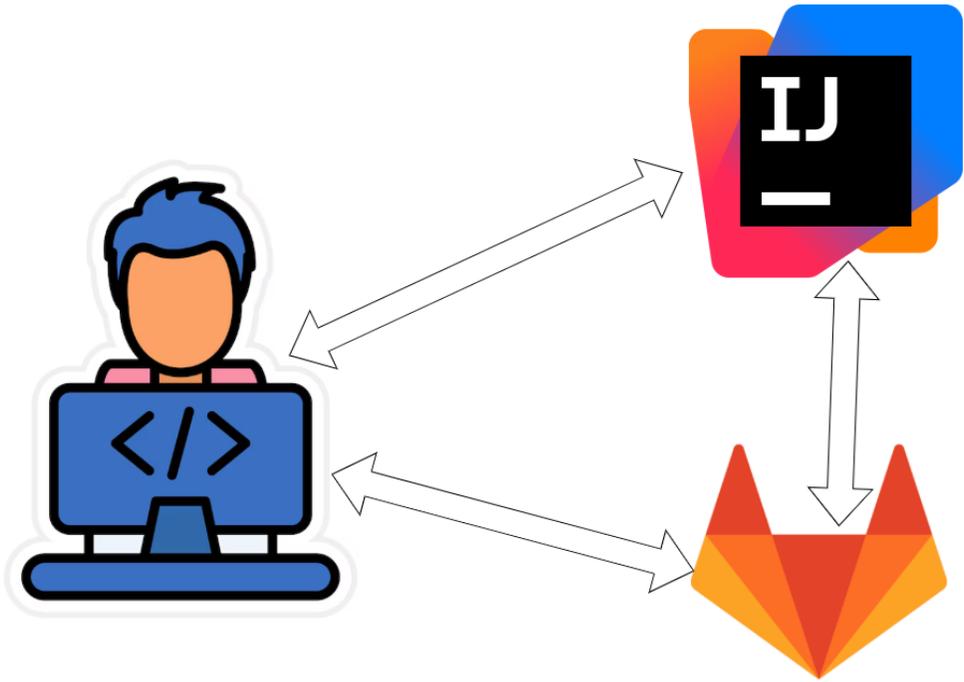
- [Git-Bob: AI Assistant for code development in data science](#)
- [OpenHands Github & Gitlab Issue Resolver](#)
- [CodeRabbit for VS Code Is Here — Free AI Code Reviews](#)
- [Augment Code: Remote Agent clears your backlog while you plan what's next](#)



Путь в написании ИИ ассистента

Или как докатиться до темной стороны сервера....

Работа над проектом до ИИ

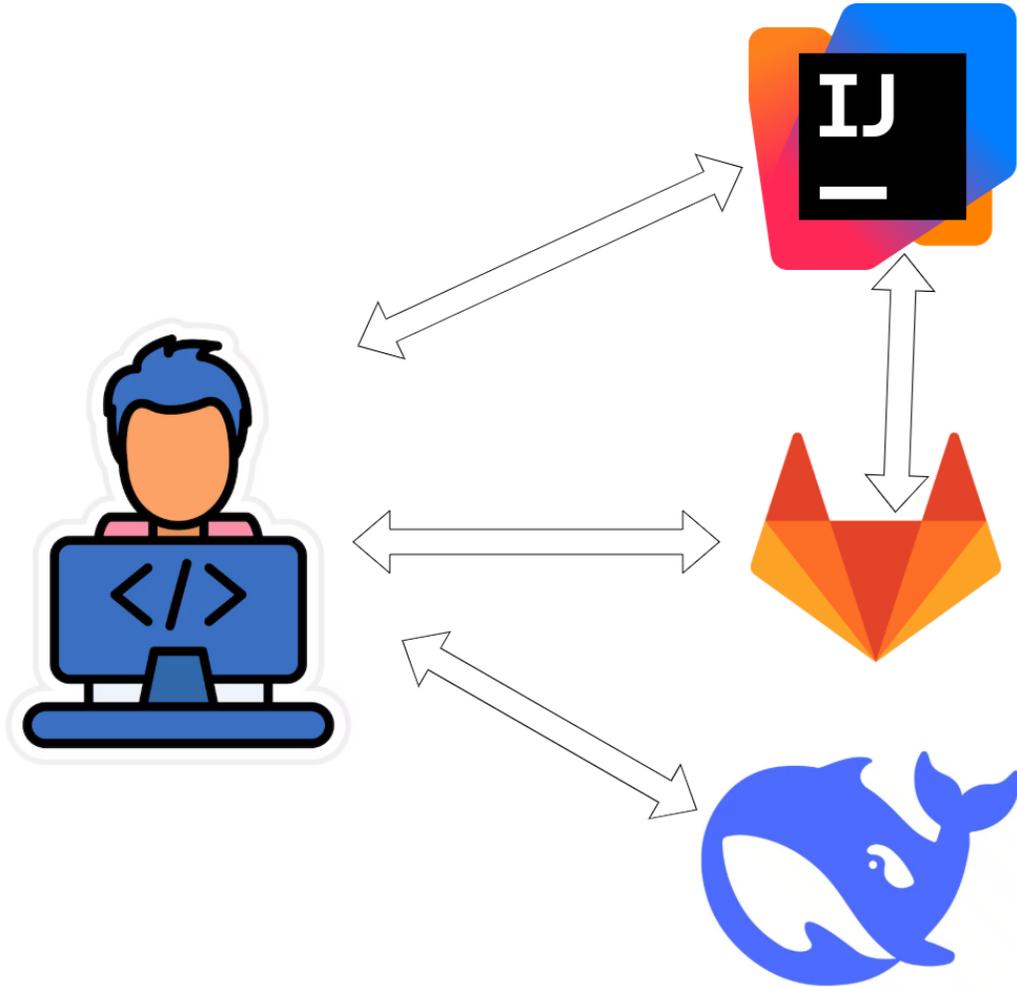


Обычный набор инструментов

IDE у вас может быть другая

Платформа для совместной разработки у вас не только может быть другой, но и отсутствовать

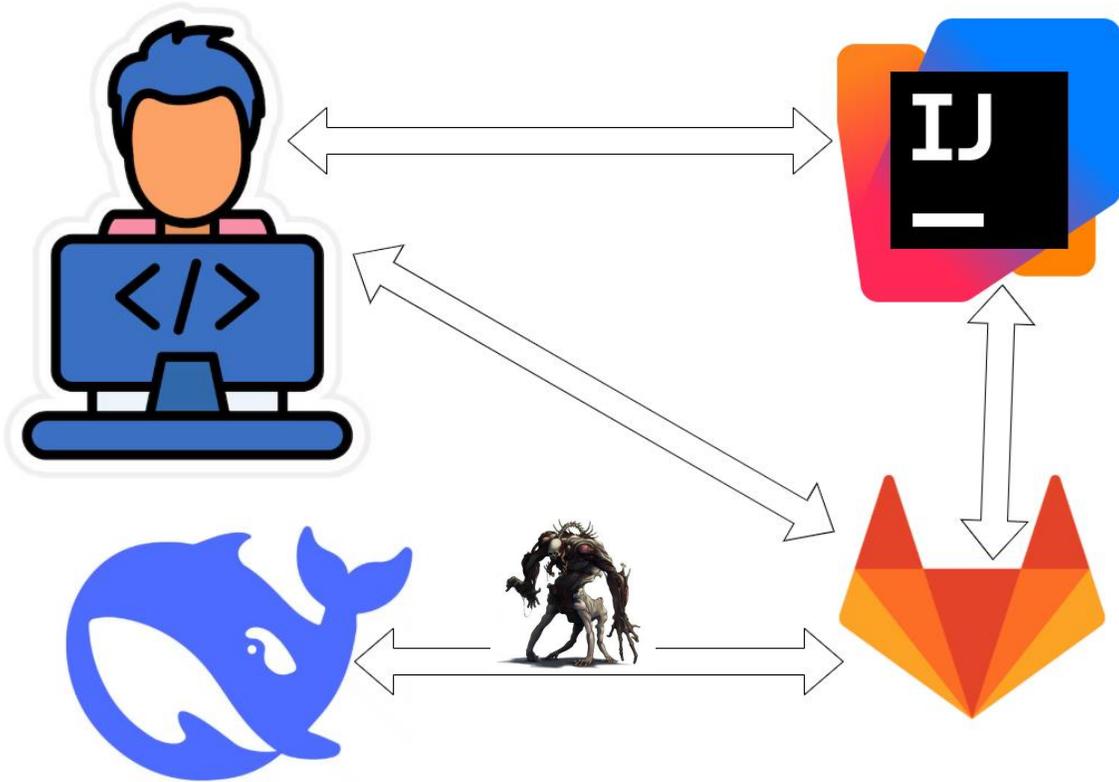
Появление ИИ



Без интеграции LLM

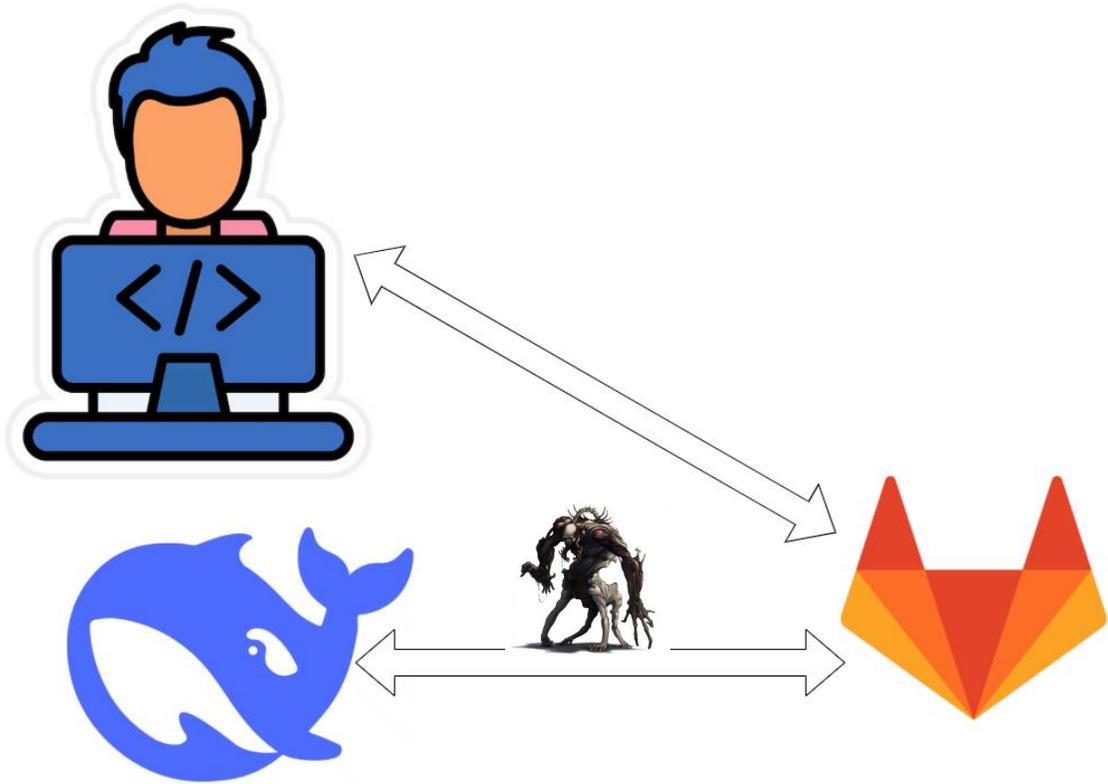
- Копировать вручную примеры кода от ИИ в IDE
- Воспроизвести код в ИИ и запросить что то в нем поменять
- Передать код и ошибки из лога сборки, чтобы ИИ их исправил
- Передать код, тест и лог падения теста, чтобы ИИ исправил код или тест

Случайно написал ассистента (Кадавр)



- опыт работы с API GitLab
- бесплатность DeepSeek
- большое количество входных токенов у DeepSeek
- маленькие проекты целиком помещающиеся в контекст

А Скрипач не нужен, родной... (с)



Если посмотреть на создание кода не глазами разработчика, а аналитика, тестировщика или руководителя разработки, то разработчик со своей IDE и помощниками ИИ мало чем отличается от ИИ, взаимодействующими со средой разработки кода напрямую.

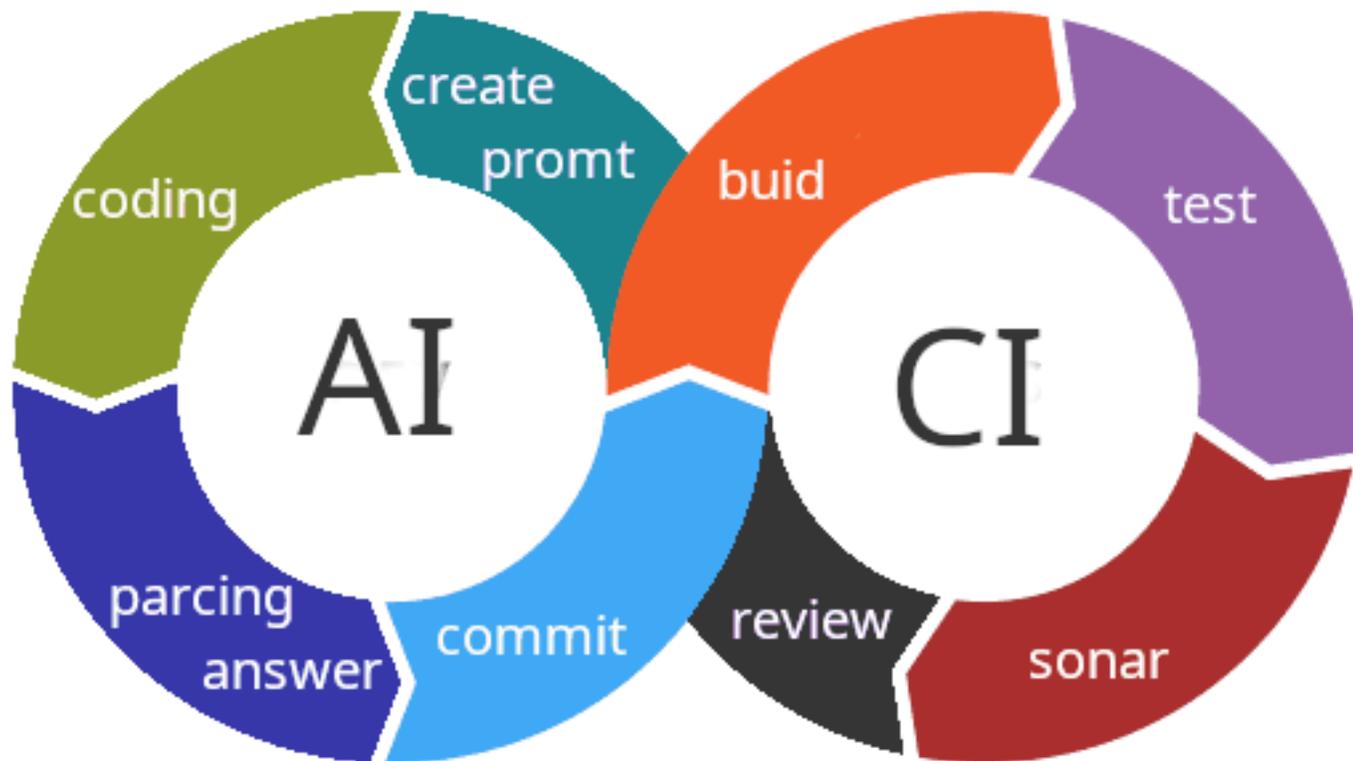
Работы в рамках запросов на изменение

- Поиск назначенных запросов на изменение
- Если pipeline в упал
 - получение логов со всех упавших job
 - выделение из лога информации о причине падения
- Если есть замечания в SonarQube
 - получение списка issue на данный запрос на изменение
 - получение не охваченных юнит тестами строк кода
- Если есть замечания по коду во время ревью
 - получение списка замечаний
- Получение issue на который реализует данный MR
- Далее как но без создания MR, а добавление commit в существующий



Цикл AI / CI 😊

ран



Кадавр и DeepSeek

- создает prompt
- генерирует ответ
- парсит ответ
- создает commit

Gitlab

- запускает pipeline
 - сборка
 - тестирование
 - анализ кода
- предоставляет
 - инструмент для ревью

Попытка покрыть тестами 100% кода



Очень высокая сложность тестов в некоторых случаях

Вал ошибок, с нескольких падающих тестов

Попытка исправить все тесты сразу усугубляла ситуацию

Решение: в промте явное указание “добавь один тест”

Некоторые ошибки сонар DeepSeek не понимает



Если в Mosquito один из аргументов к примеру `any()`, а другие аргументы константы, то константы надо обернуть в `eq()`

Если `sonar` лишний `eq()`, когда можно без него, эта ошибка

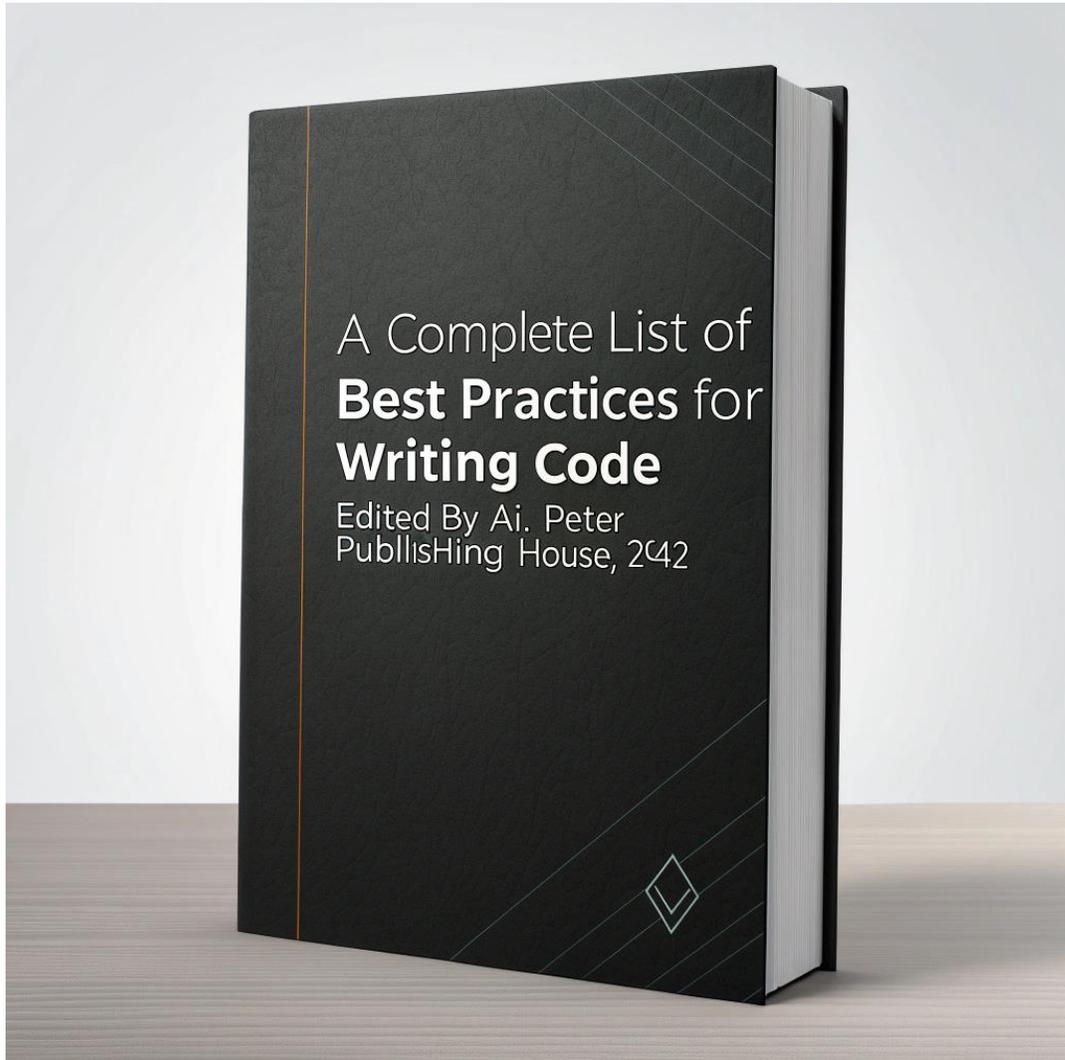
DeepSeek в обоих случаях, по аналогии правит и там где нет ошибки, итог - война правок.

Решение: в случае некоторых issue от `sonar` в `prompt` добавляются уточнения

mosquito



Пишешь новый тест, удали все остальные...



У ИИ плохо получается точно пересказывать куски кода которые не меняются.

Из за этого он часто удаляет части которые не менялись, если с кодом такое как правило приводит к падению тестов, то исчезновени тестов, комментариев приходится отслеживать на стадии ревью.

Частичное решение: каждый тест в отдельном файле

ИИ считает хорошо, но только до трех...

```
+2 -3 src/attributes/classes.js
@@ -1,10 +1,9 @@
1 1 define([
2 2   "../core",
3 3   "../var/rnotwhite",
4 4 -  "../var/strundefined",
5 5   "../data/var/dataPriv",
6 6   "../core/init"
7 7 -], function( jQuery, rnotwhite, strundefined, dataPriv ) {
+6 +], function( jQuery, rnotwhite, dataPriv ) {
8 8
9 9   var rclass = /[t\r\n\f]/g;
10 10
@@ -128,7 +127,7 @@ jQuery.fn.extend({
128 127   }
129 128
130 129   // Toggle whole class name
131 -   } else if ( type === strundefined || type === "boolean" ) {
130 +   } else if ( value === undefined || type === "boolean" ) {
132 131     if ( this.className ) {
133 132       // store className if set
134 133       dataPriv.set( this, "__className__", this.className );
```

Unified Diff Format формат описывающий изменения между двумя файлами

Существует множество инструментов умеющих их формировать и применять

ИИ прекрасно умеет его читать

Кадавр при работе с MR передает в ИИ исходный файл, и изменения в формате UDF

Но я так и не смог заставить ИИ правильно сформировать изменения в UDF

```
linuxtldr@linux:~$ diff -u file1.txt file2.txt
--- file1.txt 2023-04-18 10:22:24.964353976 +0530
+++ file2.txt 2023-04-18 10:22:52.479690702 +0530
@@ -1,4 +1,6 @@
 C
+C++
+C#
+Javascript
 Python
-Typescript
-Golang
+Golang Rust
linuxtldr@linux:~$
```

Конфликты между запросами на слияние

```
index.html — carparts-website_conflict (git: main)
13 <div id="navigation">
14 <ul>
15 <<<<<<< HEAD
16 <li><a href="index.html">Home</a></li>
17 <li><a href="about.html">About Us</a></li>
18 <li><a href="product.html">Product</a></li>
19 <li><a href="imprint.html">Imprint</a></li>
20 =====
21 <li><a href="returns.html">Returns</a></li>
22 <li><a href="faq.html">FAQ</a></li>
23 >>>>>>> develop
24 </ul>
25 </div>
```

При работе над несколькими задачами параллельно, конфликты в merge request неизбежны.

Оказывается DeepSeek не только понимает формат git для файлов в конфликте, но и довольно приемлемо умеет их решать.

В тех случаях когда ошибается, спасает цикл AI/CI и review



Ограничение на количество токенов

Pricing Details

MODEL ⁽¹⁾		deepseek-chat	deepseek-reasoner
CONTEXT LENGTH		64K	64K
MAX COT TOKENS ⁽²⁾		-	32K
MAX OUTPUT TOKENS ⁽³⁾		8K	8K
STANDARD PRICE (UTC 00:30-16:30)	1M TOKENS INPUT (CACHE HIT) ⁽⁴⁾	\$0.07	\$0.14
	1M TOKENS INPUT (CACHE MISS)	\$0.27	\$0.55
	1M TOKENS OUTPUT ⁽⁵⁾	\$1.10	\$2.19
DISCOUNT PRICE ⁽⁶⁾ (UTC 16:30-00:30)	1M TOKENS INPUT (CACHE HIT)	\$0.035 (50% OFF)	\$0.035 (75% OFF)
	1M TOKENS INPUT (CACHE MISS)	\$0.135 (50% OFF)	\$0.135 (75% OFF)
	1M TOKENS OUTPUT	\$0.550 (50% OFF)	\$0.550 (75% OFF)

Рано или поздно подход с передачей всех файлов проекта в запрос к ИИ должен был упереться в ограничения.

Файл описания проекта

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://github.com/povolzhye/code2ai/.code2ai/project-context.schema.json",
  "title": "Code2AI Project Context",
  "description": "Контекст проекта для интеграции с ИИ-сервисами",
  "type": "object",
  "required": ["project", "groups", "files"],
  "properties": {
    "project": {
      "type": "object",
      "required": ["goal", "technologies", "languages"],
      "properties": {
        "goal": {
          "type": "string",
          "description": "Цель проекта: автоматизация взаимодействия с AI-сервисами через веб-интерфейс"
        },
        "technologies": {
          "type": "array",
          "items": {
            "type": "string"
          },
          "description": "Основные технологии проекта"
        },
        "languages": { "type": "array" },
        "endpoints": { "type": "array" }
      }
    },
    "groups": {
      "type": "array",
      "description": "Логические группы компонентов системы",
      "items": { "type": "object" }
    },
    "files": {
      "type": "array",
      "description": "Метаданные файлов проекта",
      "items": {
        "type": "object",
        "required": ["path", "description"],
        "properties": {
          "path": {
            "type": "string"
          }
        }
      }
    }
  }
}
```

json схема

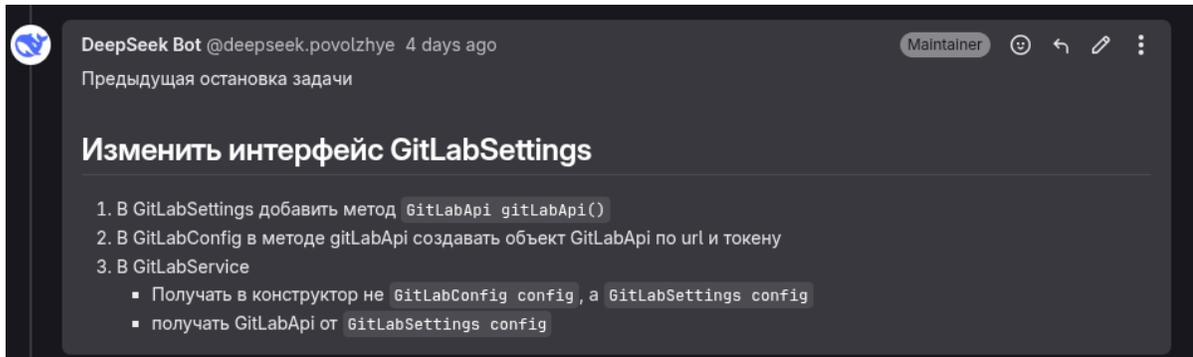
```
{
  "project": {
    "technologies": [
      "GitLab4J",
      "Gson",
      "Java",
      "JGit",
      "JUnit 5",
      "Mockito",
      "SLF4J"
    ],
    "endpoints": [
      {
        "name": "ChatService API",
        "path": "src/main/java/su/povolzhye/ai/code/ChatService.java"
      },
      { "name": "DeepSeek API" },
      { "name": "GitLabService API" }
    ],
    "goal": "Автоматизация взаимодействия с AI-сервисами через веб-интерфейсы платформ коллективной",
    "languages": [
      "English",
      "Java",
      "Russian"
    ]
  },
  "groups": [
    {
      "name": "Валидация",
      "description": "Проверка корректности конфигурации",
      "files": [
        "src/main/java/su/povolzhye/ai/code/context/ProjectContextValidator.java",
        "src/main/java/su/povolzhye/ai/code/validation/ProjectContextValidator.java",
        "src/main/java/su/povolzhye/ai/code/validation/ProjectFilesValidator.java",
        "src/test/java/su/povolzhye/ai/code/context/ProjectContextValidatorNullSchemaStreamTest.java",
        "src/test/java/su/povolzhye/ai/code/context/ProjectContextValidatorSchemaStreamNullTest.java"
      ]
    },
    {
      "name": "Инфраструктура",
      "description": "Тесты и конфигурация сборки",
    }
  ]
}
```

Описание

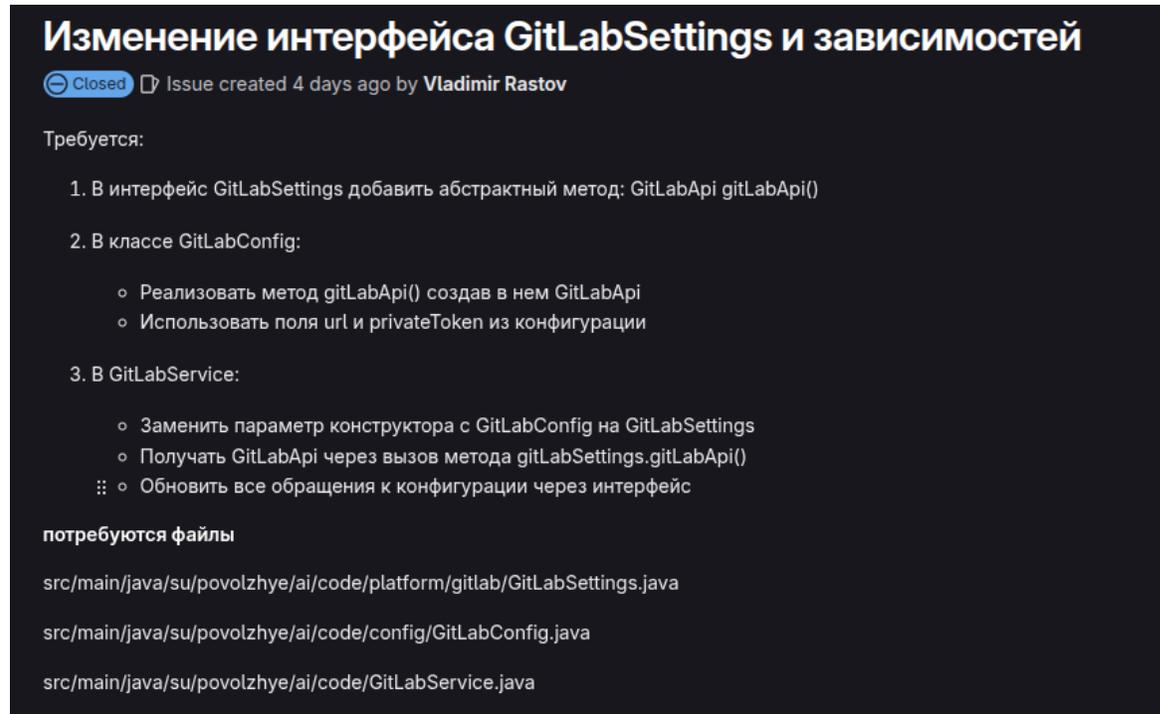
CLI для файла описания проекта

```
116 question.ai( line: "# Задание");
117 question.ai( line: "## Необходимо дополнить описание проекта используя следующий CLI");
118 question.ai( line: ""
119     ...
120     set project goal <text>                # Обновить цель проекта
121     add project technology <technology_name> # Добавить технологию
122     remove project technology <technology_name> # Удалить технологию
123     add project language <language_name>      # Добавить язык
124     remove project language <language_name>   # Удалить язык
125     add endpoint <name> <path>                # Добавить endpoint
126     update endpoint <name> path <new_path>    # Обновить путь endpoint
127     remove endpoint <name>                   # Удалить endpoint
128
129     create group <name> "<description>"        # Создать новую группу
130     update group <name> description "<text>"   # Обновить описание группы
131     tag group <name> add <tag>                 # Добавить тег к группе
132     tag group <name> remove <tag>             # Удалить тег группы
133     attach diagram <group> <path>            # Прикрепить диаграмму
134     detach diagram <group> <path>           # Удалить диаграмму
135
136     register file <path> "<description>"      # Зарегистрировать новый файл
137     deregister file <path>                   # Удалить метаданные файла
138     update file <path> description "<text>"   # Обновить описание файла
139     tag file <path> add <tag>                 # Добавить тег к файлу
140     tag file <path> remove <tag>             # Удалить тег файла
141     link file <path> dependency <dep_path>    # Добавить зависимость
142     unlink file <path> dependency <dep_path> # Удалить зависимость
```

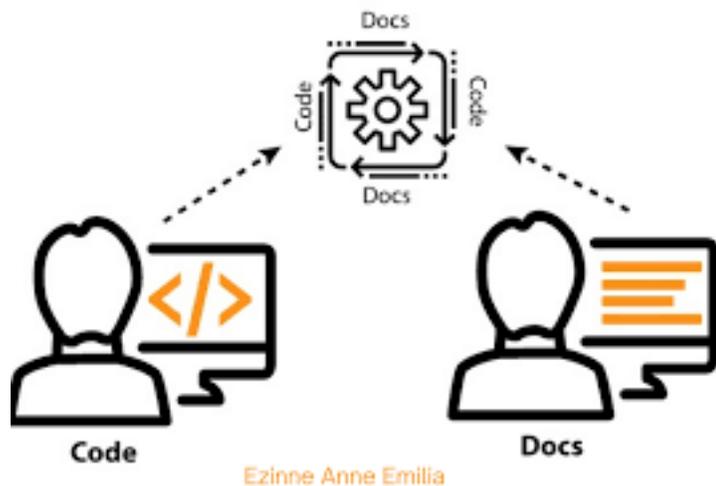
Ревью issue ИИ перед тем как писать код



Прежде чем приступать к решению задачи не плохо понять, а правильно ли вы ее поняли.



То что не работало с людьми возможно заработает с ИИ



Практику вести документацию в том же репозитории, что код, и править в одном commit, чтоб иметь консистентную коду документацию в каждом коммите, в обычном коллективе внедрить практически невозможно, или очень сложно.

ИИ прекрасно переключается между видами деятельности. Скооперировать в одном месте и времени два ИИ гораздо проще, чем двух человек.

Определить границы



Что лучше делает человек, что ИИ, а что алгоритмы.

Статический анализ кода, алгоритмически.

Разработка нового, чему не обучался ИИ поручить человеку.

Все что ИИ может сделать за приемлемое время и деньги приемлемого качества, поручать ИИ.



Спасибо за внимание!

Будем рады вашим вопросам!

[t.me / aitsvet](https://t.me/aitsvet)

[t.me / code2ai](https://t.me/code2ai)

