



RuStore

Михаил Емельянов

Директор департамента разработки

Personal AI Workspace: как делиться скиллами, шаблонами и AGENTS.md между командами



О себе

Михаил Емельянов

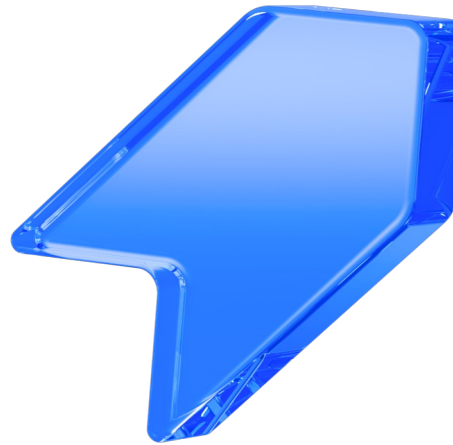
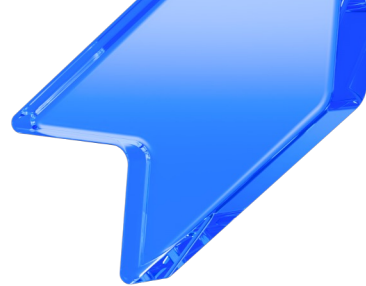
Директор департамента разработки
RuStore

- 15+ лет в разработке
- 8+ лет управления командами
- Внедряю AI в процессы разработки
- Веду Telegram-канал @ai_adopter



О чём поговорим

Проблема	Контекст между сессиями, проектами и людьми
Идея	Общий слой знаний и практик
Устройство	Архитектура, System, Shared, Personal
Примеры	Поиск, Android-задача, MCP Recover
Выводы	Польза, распространение и стоимость сопровождения
Как сделать у себя	Минимальный переносимый контекст





01 Проблема

Новая задача.
Какой контекст получит агент?

Контекст между сессиями и командами

Новая задача требует не только кода репозитория. Агенту нужны сведения об инженере, проекте и принятых способах работы.

Профиль инженера

Роль, принципы, стиль общения и зона ответственности.

Связи проектов

Задачи, решения и зависимости за пределами 1 репозитория.

Практики команды

Ролевые навыки, инструкции, шаблоны и порядок проверки результата.

Опыт прошлых задач

Что уже пробовали, почему ошиблись и при каких условиях сработало решение.

Если эти знания остаются у 1 инженера и только в рамках 1 сессии, коллеги собирают их заново.
Нужны **общий источник и способ доставки**.

Готовые механизмы уже есть

ПОДХОД	ЧТО ДАЁТ	ЧТО НУЖНО РЕШИТЬ ДОПОЛНИТЕЛЬНО
Claude Code: CLAUDE.md и память	Инструкции проекта, личные настройки, опыт между сессиями	Проверять актуальность и перенос в другой агент
Cursor: Project / User / Team Rules	Правила проекта, инженера и команды	Настраивать применение и поддерживать правила при смене клиента
Общий Git-репозиторий знаний	Версии, ревью, навыки и шаблоны команды	Организовать доставку, подключение и отделить личное

Наша задача: связать эти механизмы в переносимый слой контекста для проектов, команды и разных агентов.



02 Решение

RAIW: общий слой
практик над всеми репо

PAIW: слой контекста и практик

Личный workspace инженера: `~/.paiw/`. Профиль, контекст проектов, знания и сохранённый опыт.

Общие практики приходят из командных источников и плагинов. Адаптеры подключают их к выбранному агенту.

Подключённые проекты используют общие источники контекста и навыков. Формат подключения зависит от агента.



PARA и PAIW: разные задачи

PARA (Tiago Forte): Projects · Areas · Resources · Archives. Подход к организации материалов; PAIW добавляет подключение контекста и практик к агенту.

PARA / SECOND BRAIN

Для человека: где хранить заметки

Заметки и материалы

Структура хранения знаний

Человек обращается к материалам

Распределение по областям

PAIW

Для агента: контекст и практики

Исполняемое: skills, hooks, MCP

Слои system / shared / personal: разные уровни доступа

Агент читает подключённый контекст

Накопление знаний: learnings между сессиями

PAIW на 1 схеме

Источник обновлений

Реестр релизов

PAIW: CLI + workspace

Плагины команды

skills + общий контекст

В репозитории

.paiw/ + AGENTS.md

связь с workspace

AI Launcher / терминал

выбор · установка · обновление

paiw CLI

install · update · addon

PAIW

~/.paiw/ + подключённые плагины

System ядро и правила

Shared практики команды

Personal профиль, проекты, опыт

агент: файлы · paiw search / index

Компоненты подключает пользователь.

настроить

Инженер

Задача
и диалог

адаптеры

Контекст
skills · hooks

Агент

- Claude Code
- Codex
- OpenCode

запись опыта

репозиторий: код и MR
MCP: задачи и документы

Практика для команды: опыт → review → выпуск плагина → обновление у коллег.

Инженер: 2 входа

Источник обновлений

Реестр релизов

PAIW: CLI + workspace

Плагины команды

skills + общий контекст

В репозитории

`~/.paiw/ + AGENTS.md`

связь с workspace

Компоненты подключает пользователь.

AI Launcher / терминал

выбор · установка · обновление

paiw CLI

install · update · addon

PAIW

`~/.paiw/` + подключённые плагины

System ядро и правила

Shared практики команды

Personal профиль, проекты, опыт

агент: файлы · `paiw search / index`

настроить

Инженер

Задача
и диалог

Агент

- Claude Code
- Codex
- OpenCode

адаптеры

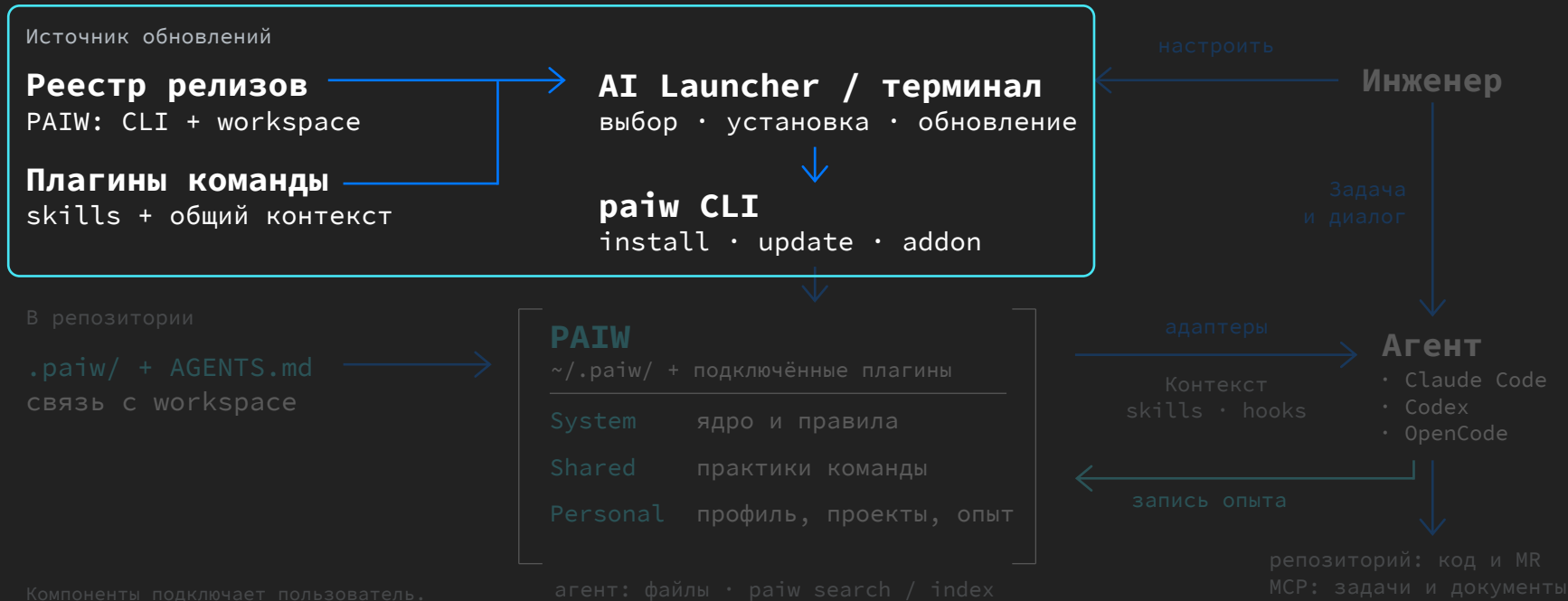
Контекст
skills · hooks

запись опыта

репозиторий: код и MR
MCP: задачи и документы

Практика для команды: опыт → review → выпуск плагина → обновление у коллег.

Откуда приходят обновления



Практика для команды: опыт → review → выпуск плагина → обновление у коллег.

PAIW: 3 области

Источник обновлений

Реестр релизов

PAIW: CLI + workspace

Плагины команды

skills + общий контекст

В репозитории

`~/.paiw/ + AGENTS.md`

связь с workspace

Компоненты подключает пользователь.

AI Launcher / терминал

выбор · установка · обновление

paiw CLI

install · update · addon

PAIW

`~/.paiw/ + подключённые плагины`

System ядро и правила

Shared практики команды

Personal профиль, проекты, опыт

агент: файлы · `paiw search / index`

настроить

Инженер

Задача
и диалог

адаптеры

Контекст
skills · hooks

Агент

- Claude Code
- Codex
- OpenCode

запись опыта

репозиторий: код и MR
MCP: задачи и документы

Практика для команды: опыт → review → выпуск плагина → обновление у коллег.

Как подклюён репозиторий

Источник обновлений

Реестр релизов

PAIW: CLI + workspace

Плагины команды

skills + общий контекст

В репозитории

`.paiw/ + AGENTS.md`
связь с workspace

AI Launcher / терминал

выбор · установка · обновление

paiw CLI

install · update · addon

PAIW

~/.paiw/ + подклюённые плагины

System	ядро и правила
Shared	практики команды
Personal	профиль, проекты, опыт

агент: файлы · `paiw search / index`

настроить

Инженер

Задача
и диалог

адаптеры

Контекст
skills · hooks

Агент

- Claude Code
- Codex
- OpenCode

запись опыта

репозиторий: код и MR
MCP: задачи и документы

Компоненты подклюёт пользователь.

Практика для команды: опыт → review → выпуск плагина → обновление у коллег.

Как агент получает контекст

Источник обновлений

Реестр релизов

PAIW: CLI + workspace

Плагины команды

skills + общий контекст

В репозитории

`~/.paiw/ + AGENTS.md`

связь с workspace

Компоненты подключает пользователь.

AI Launcher / терминал

выбор · установка · обновление

paiw CLI

install · update · addon

PAIW

`~/.paiw/` + подключённые плагины

System ядро и правила

Shared практики команды

Personal профиль, проекты, опыт

агент: файлы · `paiw search / index`

настроить

Инженер

Задача
и диалог

адаптеры

Контекст
skills · hooks

Агент

- Claude Code
- Codex
- OpenCode

запись опыта

репозиторий: код и MR
MCP: задачи и документы

Практика для команды: опыт → review → выпуск плагина → обновление у коллег.

Что делает агент

Источник обновлений

Реестр релизов
PAIW: CLI + workspace

Плагины команды
skills + общий контекст

В репозитории

`~/.paiw/ + AGENTS.md`
связь с workspace

Компоненты подключает пользователь.

AI Launcher / терминал
выбор · установка · обновление

paiw CLI
install · update · addon

PAIW

`~/.paiw/ + подключённые плагины`

System ядро и правила

Shared практики команды

Personal профиль, проекты, опыт

агент: файлы · `paiw search / index`

настроить

Инженер

Задача
и диалог

адаптеры

Контекст
skills · hooks

Агент

- Claude Code
- Codex
- OpenCode

запись опыта

репозиторий: код и MR
MCP: задачи и документы

Практика для команды: опыт → review → выпуск плагина → обновление у коллег.

Как опыт становится практикой

Источник обновлений

Реестр релизов

PAIW: CLI + workspace

Плагины команды

skills + общий контекст

В репозитории

`~/.paiw/ + AGENTS.md`

связь с workspace

AI Launcher / терминал

выбор · установка · обновление

paiw CLI

install · update · addon

PAIW

`~/.paiw/ + подключённые плагины`

System ядро и правила

Shared практики команды

Personal профиль, проекты, опыт

агент: файлы · `paiw search / index`

настроить

Инженер

Задача
и диалог

адаптеры

Контекст
skills · hooks

Агент

- Claude Code
- Codex
- OpenCode

запись опыта

репозиторий: код и MR
MCP: задачи и документы

Компоненты подключает пользователь.

Практика для команды: опыт → review → выпуск плагина → обновление у коллег.

System: общий механизм работы

Загрузка контекста

Точка входа и адаптеры связывают workspace с агентом. Формат подключения зависит от клиента.

Правила работы

Где искать знания, куда сохранять результат и какие каталоги управляются поставкой.

Базовые навыки

Подключение проекта, поиск, работа со знаниями. Это функции ядра PAIW.

Обновление

CLI устанавливает и обновляет ядро и адаптеры. Личные данные хранятся отдельно.

Далее разберём CLI и hooks. После них: что команда добавляет в Shared.

System: установка и подключение

КОМАНДА	ЧТО ДЕЛАЕТ
<code>paiw install / update</code>	Устанавливает или обновляет workspace и адаптеры
<code>paiw item enable <id></code>	Подключает выбранный навык, агента или команду
<code>paiw export personal</code>	Обновляет уже подключённые личные компоненты
<code>paiw index / search</code>	Индексирует знания и ищет нужные источники
<code>paiw validate</code>	Проверяет структуру workspace

AI Launcher вызывает CLI при установке и обновлении. Поиск и работу с файлами может выполнять агент.

System: действия в нужный момент

Хук	Когда	Что делает
session-start	старт сессии	передаёт профиль и контекст проекта
durable-write	сохранение факта	подсказывает, куда положить (project / global)
write-guard	запись файла	ограничивает запись в */shared/

События и возможности hooks зависят от клиента и подключённых компонентов.

Хуки Claude Code (общий слой Paiv)

Скрипты для событий **Claude Code Hooks** <https://code.claude.com/docs/en/hooks>: в **settings.json** указывайте абсолютные пути к файлам в вашем checkout этого репозитория.

Канонические метаданные (catalog v1)

Файлы ***.sh** содержат встроенный блок **PAIW_HOOK_CATALOG** (heredoc), как показано в **paiv/docs/catalog-item-v1.md**. JS-хуки (***.js** / ***.cjs** / ***.mjs**, включая **send-ai-logs.mjs**) описывают метаданные через **const paivMeta = { ... }** в самом runtime-файле; объект парсится статически (без выполнения JS), как и у OpenCode plugin runtimes.

Скрипты

Файл	Событие (типично)	Назначение
notify-stop.sh	Stop	macOS: уведомление о конце ответа. Env: NOTIFY_STOP_TITLE , NOTIFY_STOP_TERMINAL_APP ; корень проекта — CLAUDE_PROJECT_DIR (задаёт Claude). В stdout отдаёт JSON с systemMessage о затронутом файле.
after-tool-call.sh	PostToolUse (Write, Edit, Read, NotebookEdit)	В stdout отдаёт JSON с systemMessage о затронутом файле.
paiv-write-guard.sh	SessionStart , PreToolUse , PostToolUse (см. paiv install)	Задаёт paiv guard hook — event "\$1" — в PATH должен быть бинарник paiv .
send-ai-logs.mjs	PostToolUse / Bash	Отправка диффа на AI_LOG_ENDPOINT ; без переменной тихо выходит. Работает только если каталог проекта внутри репозитория Git.

Shared: практика становится общей

Что публикуем

Общие инструкции, шаблоны, навыки и проверенные решения. Личные данные в командную поставку не входят.

Где меняем

В исходном репозитории практики. Изменение проходит ревью, затем поставляется в workspace или плагин.

Как подключаем

Материалы доступны в shared-разделах. Skills и агенты подключаются командными плагинами.

Как проверяем

Коллега получает нужную версию и повторяет сценарий в своём проекте. Проверяем факт применения, а не наличие файла.

Навык описывает порядок работы, MCP предоставляет доступ к системе. Следующий слайд показывает эту связку.

Skills + MCP: практика и доступ

Навыки задают порядок работы. MCP даёт доступ к рабочим системам.

КОМАНДНЫЕ

[android-explore-task](#)

[android-code-task](#)

[android-self-review-task](#)

ANDROID

план задачи

реализация

проверки перед ревью

БАЗОВЫЕ

[paiw-onboard-personal](#)

[paiw-onboard-project](#)

[paiw-knowledge-update](#)

[self-improvement](#)

PAIW

личный профиль

контекст проекта

единый источник

запись опыта

ЛИЧНЫЕ

[my-mcp-recover](#)

[my-planning-week](#)

диагностика MCP

план недели

MCP

Jira

GitLab

Confluence

YouGile

Календарь

Рабочие чаты

ДОСТУП К СИСТЕМАМ

Требования, задачи, эпикс

Код, MR и проверки CI

Документы и договорённости

Личные задачи и проекты

Встречи и свободное время

Обсуждения и поиск контекста

Подборка из установленного каталога. Командные Android-навыки поставляются плагином RuStore и подключаются под задачу.

Personal: ваш рабочий контекст

ГДЕ ХРАНИТСЯ

ЧТО СОДЕРЖИТ

[identity/personal/](#)

Роль, принципы, стиль общения

[projects/personal/](#)

Контекст проектов, решения, ссылки на репозитории

[resources/personal/](#)

Рабочие заметки, шаблоны, знания

[tools/personal/](#)

Ваши skills, агенты и команды

[.learnings/](#)

Опыт ошибок: глобальный или внутри проекта

Личный контекст остаётся у инженера. В Shared переносим только подготовленные
и проверенные общие практики.

Personal: подключение через диалог

Вы отвечаете на вопросы, агент записывает согласованные сведения в нужные файлы.

НАВЫК

СОДЕРЖАНИЕ

`paiw-onboard-personal`

Роль, цели и стиль работы

`paiw-onboard-project`

Назначение, сборка
и правила проекта

Схема диалога

Агент: роль, задачи, язык ответов?

Инженер: описывает свою работу.

Агент: уточняет источники и пути, записывает согласованный контекст.

Пример вопроса при подключении проекта

«Какой slug использовать для проекта в PAIW?»

Выбранное имя связывает репозиторий с `projects/personal/<slug>/`.

После подключения проверьте, что агент находит профиль и сведения о нужном проекте.

Что агент получает на старте

Реальная новая сессия Claude Code в репозитории этой презентации.

01 · Sessionstart

Hook PAIW запускается перед ответом агента.

02 · Bootstrap

CLI определяет проект и возвращает указатели.

03 · Контекст сессии

Агент видит эти данные до чтения файлов.

Сокращённый журнал фактического запуска

```
SessionStart:startup success · exit 0 slug: paiw-  
session has_slug: true recent_files:  
README.md, notes.md, status.md, tasks.md,  
repo tools: []
```

Агент назвал проект и файлы при отключённых инструментах. На старте переданы указатели.
Полный текст документов нужно читать отдельно.

Personal: 1 факт, 1 источник

ЧТО СОХРАНЯЕМ

ГДЕ ХРАНИТСЯ

Профиль и стиль работы

identity/personal/

Контекст и решения проекта

projects/personal/<slug>/

Повторно используемое знание

resources/personal/

Опыт ошибки

Глобальная или проектная .learnings/

Рабочий документ или отчёт

В самом проекте, без 2-й копии в памяти

SSOT: полный факт хранится в 1 месте. Остальные документы ссылаются на него.

2 инструкции противоречат друг другу

Учебный пример: команда изменила способ запуска тестов, а старые заметки остались.

До: 2 ответа

В памятке проекта: `testDebugUnitTest`.

В старой заметке: `test`.

Агент может выбрать не ту команду.

Действие навыка

Найти оба упоминания. Проверить актуальную команду по репозиторию и уточнить неоднозначность у владельца.

Устранить конфликт.

После: 1 источник

Рабочая команда записана в `docs/testing.md` проекта. В остальных заметках остаются ссылки на этот файл.

Проверка

В новой сессии спросить, как запустить тесты. Проверить команду и файл, на который ссылается агент.

Польза: при изменении практики обновляется 1 источник.
Конфликтующие копии не остаются незаметными.

Personal: как сохраняется опыт

`~/.paiw/.learnings/`

← знания по всей структуре

|— ERRORS.md |— LEARNINGS.md |— FEATURE_REQUESTS.md

`~/.paiw/projects/personal/<slug>/.learnings/`

← знания этого проекта

Навык **self-improvement** помогает агенту сохранить опыт в конце задачи по правилам

Для команды: отобрать полезное, проверить и опубликовать в общем источнике. Затем проверить применение у коллеги.

Запись сохраняет опыт. Применение проверяем в новой сессии.

Как проверить применение памяти

Новая сессия Claude Code. Указаны навык MCP Recover и журнал опыта; причина сбоя и нужное правило в запрос не переданы.

01 · сохранённый разбор

Horizon BI отклонял авторизацию.
Правило сохранено в навыке и [ERRORS.md](#).

ERR-20260906-001

02 · повторная задача

Агент прочитал навык, нашёл разбор
и запустил mcp-health.sh.

Horizon BI: HTTP 403

03 · правило в ответе

Выбрал смену токена с обновлением копии
в Launcher. Учёл запрет OAuth из прежнего
разбора.

04 · что подтверждено

В журнале есть чтение источников, проверка сервера и
ссылка на правило. Настройки подключений не изменены.

[Правило использовано при диагностике](#). Восстановление доступа требует нового токена и отдельной проверки. Это 1 наблюдаемый запуск.



04 Примеры

Что реально помогает
каждый день

3 части повседневной работы

СЦЕНАРИЙ

ЧТО ПОКАЗЫВАЕТ

01 · Найти контекст

`pa1w search`: получить ссылки на нужные знания

02 · Реализация Android-задачи

Общий набор навыков: исследование, реализация, проверка

03 · Разобрать сбой MCP

Навык диагностики использует сохранённый опыт

Общая последовательность:
найти источник → применить практику → проверить результат.

Поиск возвращает источники

Пути из реальной JSON-выдачи, без локального префикса.

```
$ paiw search session-start --path-prefix resources/shared/agent-tooling --limit 3 --json
```

resources/shared/agent-tooling/write-system.md

индекс

resources/shared/README.md

связанный документ

resources/shared/agent-tooling/overview.md

связанный документ

Поиск находит исходные и связанные материалы. Агент читает источник и проверяет его применимость к задаче.

Командные навыки для Android-разработки

Подборка из установленного плагина RuStore: от требований до проверки приложения.

НАВЫК	РЕЗУЛЬТАТ
android-explore-epic	Исследовать требования эпика и кодовую базу
android-decompose-epic	Подготовить техдизайн и разбиение на задачи
android-explore-task	Проанализировать задачу и составить план
android-code-task	Реализовать подготовленный план
android-self-review-task	Сверить код с планом; build, lint, detekt, unit-тесты
generate-android-test-cases	Подготовить тест-кейсы для критичных сценариев
android-ui-automation	Проверить сценарии и навигацию через ADB
android-build-apk	Собрать APK с выбранной версией и вариантом

Основной путь задачи: [explore-task](#) → [code-task](#) → [self-review-task](#).

Для UI-проверок дополнительно нужны устройство или эмулятор и контекст приложения.

Сначала определить причину сбоя

Реальный запуск [mcp-health.sh](#) по инструкции навыка [paiw-my-mcp-recover](#).

Проверить работу через MCP

В отдельной проверке Jira ответила и на живой вызов MCP. Снимок показывает диагностику, а не восстановление после сбоя.

Разделить причины

Horizon BI и Miro отклонили авторизацию. Нужно проверить их учётные данные. Рабочие токены других сервисов менять незачем.

```
--- generic:confluence
OK бинарь на месте
OK auth OK (HTTP 200)
--- generic:gitlab
OK бинарь на месте
OK auth OK (HTTP 200)
--- generic:jira
OK бинарь на месте
OK auth OK (HTTP 200)
```

Horizon BI: отказ авторизации

```
FAIL token в headers протух (HTTP 403 invalid credentials).
```

Снимок WezTerm. Выбраны строки без корпоративных адресов.

Проверка памяти: новая сессия применила сохранённый разбор Horizon BI и выбрала обновление токена с учётом Launcher. Восстановление доступа ещё требует нового токена.

Что остаётся после завершения задачи

Контекст

Ссылка на проверенный источник. Следующая сессия может найти нужный факт без пересказа всей истории.

Практика

Навык с входными условиями, порядком действий и проверкой результата. Его можно применить в другом проекте.

Опыт ошибки

Условие, при котором сработало решение.
Новая сессия показывает, применяется ли правило на практике.

Критерий пользы

Другой инженер повторяет сценарий без автора.
Сравниваем результат и количество ручных подсказок.



05 Итоги

Что это дало
и как сделать у себя

Как инструменты попадают к инженеру

AI Launcher: фрагмент экрана «Мои инструменты». Здесь видны версии, обновления и состояние MCP.

Все 31			Обновления 0	Нужна настройка 0	Для всех агентов ⓘ	
Инструмент			Версия		Состояние	
	confluence Confluence: поиск и управление страницами в Data Center. Получить токен: https://confl...	generic	2.2.3		☒ Вкл.	
	gitlab GitLab: проекты, merge requests, коммиты и code review. Получить токен: https://gitlab.c...	generic	2.2.3		☒ Вкл.	

Доставка

Установка и обновление агентов, плагинов, MCP и PAIW.

Рабочий агент

Claude Code, Codex, OpenCode и другие клиенты через адаптеры.

Роль PAIW

Профиль, проекты и память. Общие навыки поставляются плагинами.

Безопасность

LLM Proxy, защитные механизмы, политики и лимиты доступа.

Подход переносим: доставка контекста и настройка агента доступны без внутренней инфраструктуры RuStore.

Что меняется в работе

ПОВТОРЯЮЩАЯСЯ ЗАДАЧА

Передать контекст проекта

Вернуться к прошлой ошибке

Применить практику в другом репозитории

Найти прежнее решение

МЕХАНИЗМ PAIW

Подключить профиль и сведения о проекте через workspace

Найти запись в .learnings/ и проверить правило

Подключить тот же навык и материалы

Получить ссылки через paiw search

Эффект оцениваем по ручным подсказкам, исправлениям и повторному применению. Само наличие файлов его не доказывает.

Что нужно для общей практики

Личный результат из предыдущего слайда ещё нужно сделать воспроизводимым для коллег.

Профиль инженера

У практики есть место хранения и ответственный за актуальность. Личный контекст остаётся отдельно.

Связи проектов

Пример использования и ревью перед публикацией. Неудачный результат тоже фиксируется.

Практики команды

Коллеги получают согласованную версию навыка и его материалов. Можно понять, какая версия подключена.

Опыт прошлых задач

Новый проект и новая сессия. Агент находит практику, применяет её, а инженер проверяет результат.

Если эти знания остаются у 1 инженера, коллеги собирают их заново.
Нужны **общий источник** и **способ доставки**.

PAIW: слой контекста и практик

Нужен агент с доступом к файлам. PAIW, MCP и корпоративный launcher для этого примера не требуются.

Помогите собрать переносимый контекст для моего проекта.

1. Уточните мою роль и 1 повторяющуюся задачу команды.
2. Создайте `~/ai-context/personal/profile.md` и `~/ai-context/shared/review.md`.
Запишите профиль и шаги задачи.
3. В `AGENTS.md` текущего проекта добавьте абсолютные пути к этим файлам и правило читать их перед выполнением задачи. Существующие инструкции сохраните. Проверьте поддержку `AGENTS.md` моим агентом; при необходимости настройте его вход.
4. Выполните задачу с этим контекстом. Покажите источники и результат проверки. Не публикуйте личный профиль.



Полный текст запроса

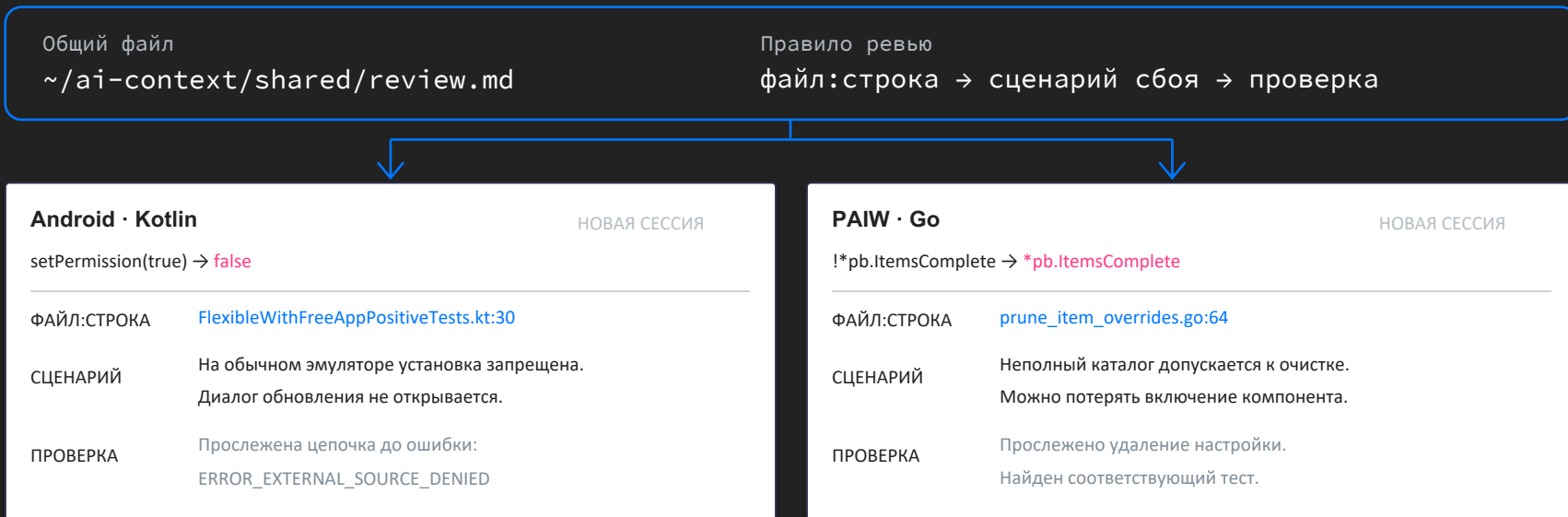


Пример на GitHub

Проверьте новую сессию и 2-й репозиторий. Для коллег отдельно опубликуйте общий файл и подключите его.

1 файл правил, 2 репозитория

Реальный тест промпта: новые сессии Codex, 2 worktree, по 1 контрольной ошибке.



Правило не повторяли в запросах. Обе сессии прочитали тот же файл и применили его.

Ответы сокращены. Контрольные изменения восстановлены. Статический анализ, без запуска сборки и тестов.

Недостатки и стоимость сопровождения

Контекст устаревает

Изменения в коде и процессах нужно переносить в знания. Противоречивые записи приводят к неверным решениям.

Лишний контекст мешает

Большой объём правил расходует окно контекста и затрудняет выбор нужного. Нужны отбор и поиск по задаче.

Доставка требует поддержки

Разные агенты читают инструкции и подключают навыки по-разному. Адаптеры и совместимость версий нужно проверять.

Память не гарантирует применение

Записанное правило агент может не найти или неверно применить. Нужны повторные проверки и сопровождение практик.

В стоимость входят настройка, обновление знаний и проверка поведения агента. Снижение этой стоимости нужно подтверждать на своих задачах.



RuStore

Михаил Емельянов

Директор департамента разработки

Вопросы

tg: @ai_adopter

