Михаил Кондратьев

AI Engineer / LLM-разработчик

Делаю AI-функции под ключ: от разговора с заказчиком и выбора технологий до дообучения модели, агента, интерфейса, сборки и выкладки. Последний проект — AI-ассистент, который работает на телефоне без интернета и управляет большой промышленной программой голосом.

@gunity_dev · mike@obb.su · ai.u-dev.ru · github.com/gunity · +7 (991) 113-34-20

ОПЫТ	ФОРМАТ	ЛОКАЦИЯ
5 лет 10 месяцев	Удалённо, полная занятость	Тверь

0,4 с — ответ AI-ассистента на телефоне, без интернета

20 тыс. — обучающих примеров, каждый проверен в программе

10M+ MAU — у продуктов, под которые писал инфраструктуру

5,5 лет — коммерческой разработки: веб, мобильные, инфраструктура

01 Чем занимаюсь

Довожу AI до работающего продукта сам: модель, агент, интерфейс, сервер, сборка и контроль качества. Решения принимаю по замерам, а не на глаз.

AI-агенты и поиск по документам

Агенты, которые сами выполняют действия в программе: вызывают инструменты, отдают структурированный ответ, просят человека подтвердить опасное. Поиск по документам с ответом и ссылкой на источник.

`AI-агенты` `tool calling` `MCP` `RAG` `LangGraph`

Дообучение и модели на устройстве

Собираю обучающие данные, дообучаю модель, сжимаю её для телефона и запускаю без интернета. Распознавание речи тоже работает на самом устройстве.

`LoRA` `MLX` `GGUF` `llama.cpp` `sherpa-onnx`

Качество и скорость

Замеры на отложенных наборах вместо «вроде работает»: точность, скорость ответа, расход токенов. По ним выбираю модель, железо и настройки.

`eval` `LLM-as-a-judge` `Langfuse` `latency`

Продукт и инфраструктура

Веб, мобильные и настольные приложения, серверная часть, сборка и выкладка без ручных шагов. Пять лет коммерческой разработки до AI — отсюда инженерная база.

`Python` `TypeScript` `Docker` `Kubernetes` `CI/CD`

02 Проекты для заказчиков

Кроме основной работы делаю AI-решения для компаний: начинаю с того, что нужно бизнесу, и подбираю решение под бюджет.

Ассистенты по документам

`RAG для базы знаний компании`

Ассистент отвечает на вопросы сотрудников по внутренним документам и показывает, откуда взят ответ.

- Разбор и нарезка документов, эмбеддинги, гибридный поиск и reranking.
- Оценка качества ответов и трейсинг, чтобы видеть, где ассистент ошибается.

`Python` `FastAPI` `LangGraph` `Qdrant` `pgvector` `Langfuse`

Модели в закрытом контуре

`Когда данные нельзя отдавать наружу`

Разворачиваю open-source модели на серверах заказчика с тем же API, что у OpenAI, — приложения переключаются без переделки.

— Сервер моделей, автоматическая выкладка и мониторинг.

vLLM Qwen Kubernetes Prometheus Grafana Terraform

Агенты для автоматизации

AI внутри рабочих процессов

Агенты, подключённые к CRM, таск-трекерам и внутренним системам компании, берут на себя рутинные операции.

— Подключение через MCP и вызов инструментов, структурированные ответы.

MCP tool calling OpenAI API GigaChat Redis

Свой код

Открытые проекты

Настольные приложения на Rust (Tauri) под Windows, macOS и Linux, проекты на TypeScript и .NET.

— Сравнивал TypeScript и Rust/WebAssembly на тяжёлом расчёте, чтобы выбирать стек по цифрам.

github.com/gunity

Rust Tauri TypeScript three.js .NET

03 Как я работаю с AI-агентами

Код по моим спецификациям пишут агенты, я проектирую, проверяю и отвечаю за результат. Так за полгода получается объём, на который обычно нужна команда.

SPEC-DRIVEN

Разработка по спецификациям

Ввёл в команде процесс: задача сначала описывается, потом агент пишет код, потом тесты и ревью. Написал первые правила и skills для Claude Code, Cursor и Codex — процесс стал стандартом команды.

НЕСКОЛЬКО АГЕНТОВ

Задачи из Jira параллельно

Несколько агентов ведут задачи одновременно, каждый в своей копии репозитория. Агент сам гоняет тесты, снимает экран «было / стало» и пишет отчёт.

КОНТРОЛЬ

Человек принимает решение

В основную ветку код попадает только после зелёных тестов и проверки человеком. Сборка, подписанное Android-приложение и тестовый стенд выкладываются автоматически.

→ Правило одно: код, который человек не видел, не сливается.

04 Опыт работы

Апрель 2026 — настоящее время

8 месяцев

• СЕЙЧАС

ГЕДОКОРП gedocorp.ru

AI Engineer / Fullstack-разработчик

Программы для инженеров, которые проверяют промышленное оборудование прямо на объекте, часто без связи.

Команда: 5 разработчиков, тимлид, PM. Работаю AI-first: код пишут агенты по моим спецификациям.

- Сделал AI-ассистента, которым инженер управляет программой голосом или текстом: «поставь штуцер Ду100 на 1200», диктовка замеров, вопросы о браке и ресурсе. 48 видов действий, опасные — только после подтверждения человеком.
- Дообучил модель под эту программу и запустил на телефоне без интернета: около 20 000 обучающих примеров, каждый проверен исполнением в программе. Ответ ускорил с 2,1 до 0,4 секунды.
- Проверял качество замерами на 87 фразах, написанных без оглядки на обучение: по ним выбрал процессор вместо видеоядра телефона — 80% верных ответов против 59%.
- Сделал голосовой ввод без интернета: распознавание речи на самом устройстве сохраняет числа и номера точек так, как их сказали. Встроил его и в планшетное приложение на Unity.
- Подключил поиск по нормативам и документации (ГОСТ, РД, руководства): ассистент отвечает со ссылкой на нужный пункт.
- Основной разработчик веб-приложения «Конструктор сосудов и аппаратов» (TypeScript, React, three.js): 3D-модель, чертежи, расчёт на прочность по ГОСТ, отчёты в PDF. ~135 тыс. строк и ~5 800 автотестов, из того же кода собирается Android-приложение.
- Ввёл в команде разработку с AI-агентами по спецификациям и сделал систему, где несколько агентов параллельно ведут задачи из Jira.
- Настроил сборку и выкладку с нуля (GitLab CI, Docker): без зелёных тестов код не сливается, Android-приложение и стенд выкладываются автоматически. Развернул и настроил Jira.
- Делаю настольные приложения на Rust (Tauri) под Windows, macOS и Linux.

`LLM` `LoRA` `MLX` `llama.cpp` `GGUF` `sherpa-onnx` `Python` `TypeScript` `React` `three.js` `Capacitor` `Unity` `Rust` `Tauri` `Docker` `GitLab CI` `Claude Code` `Codex` `Cursor`

Декабрь 2021 — Декабрь 2025

4 года 1 месяц

Zimad zimad.com

Разработчик инфраструктуры (Unity, C#)

Инфраструктура, на которой работали продукты издателя с 10M+ пользователей в месяц.

Издатель мобильных игр, 300+ сотрудников. Команда: 4–6 разработчиков, PM.

- Разработал модули внутреннего SDK: вход через разные платформы, приглашения, кланы с рейтингами, чаты, аналитику поведения пользователей.
- Сократил сетевой трафик: отправлял только изменения, кешировал, убирал повторные запросы. При такой аудитории это напрямую снижает нагрузку на серверы.
- Писал нативные модули под Android (Java / Kotlin) и iOS (Objective-C) с единым API.
- Автоматизировал сборки, публикацию пакетов и тестирование: UI-тесты и unit-тесты критичных модулей.
- Работал с серверной частью на .NET (ASP.NET Core, PostgreSQL, Docker).

`C#` `Unity` `Java/Kotlin` `Objective-C` `.NET` `PostgreSQL` `Docker` `CI/CD`

Ноябрь 2020 — Декабрь 2021

1 год 2 месяца

Allemo

Unity-разработчик, техлид

Мобильные игры для Android и iOS.

Москва. Команда из 5 разработчиков.

- Был техлидом: менторил команду, проводил код-ревью, декомпозировал задачи — прототипы стали выходить вдвое быстрее.
- С нуля ввёл общие правила: архитектуру модулей, стиль кода, договорённости. Любой разработчик мог подхватить чужой код без долгого погружения.
- Сделал модули, через которые реклама, аналитика и звук подключаются за часы, а не за дни.

`C#` `Unity` `DI` `async/await` `Git`

05 Стек

ЯДРО

Python | TypeScript | LLM | AI-агенты | RAG

	Docker
АГЕНТЫ И ПОИСК	tool calling, structured output, MCP, LangGraph, LangChain, Qdrant, pgvector, reranking
МОДЕЛИ	LoRA, MLX, PyTorch, Hugging Face, GGUF, llama.cpp, vLLM, Ollama, sherpa-onnx
КАЧЕСТВО	eval-наборы, LLM-as-a-judge, Langfuse, guardrails, prompt injection
BACKEND И WEB	FastAPI, asyncio, React, three.js, Node.js, PostgreSQL, Redis, C# / .NET
ПЛАТФОРМЫ	Android, iOS, Windows, macOS, Linux, Tauri (Rust), Capacitor, Unity
ИНФРАСТРУКТУРА	Kubernetes, GitLab CI, GitHub Actions, Nginx, Prometheus, Grafana
AI В РАЗРАБОТКЕ	Claude Code, Codex, Cursor, Copilot, spec-driven, несколько агентов

06 Ещё

ЗАКАЗЧИК

Говорю с бизнесом

Собираю требования, предлагаю варианты с компромиссами по сроку, стоимости и качеству, обосновываю выбор замерами.

ИНФРАСТРУКТУРА

Свои серверы

Держу VPN-сервис на десятке серверов: Linux, nginx, TLS, мониторинг.

ЯЗЫКИ

Английский — B2

Средне-продвинутый. Русский — родной.

ОБРАЗОВАНИЕ

2020 Тверской государственный технический университет — ИТ, программная инженерия — неоконченное высшее

2018 Колледж им. А. Н. Коняева — ИТ, программист-техник — среднее специальное

07 Контакты

Быстрее всего — в Telegram, отвечаю в течение дня.

TELEGRAM
@gunity_dev

ПОЧТА
mike@obb.su

ТЕЛЕФОН
+7 (991) 113-34-20

GITHUB
github.com/gunity

Telegram по QR

Отсканируйте код камерой телефона — на нём откроется Telegram с чатом со мной. Пригодится, если сайт открыт не на вашем устройстве.