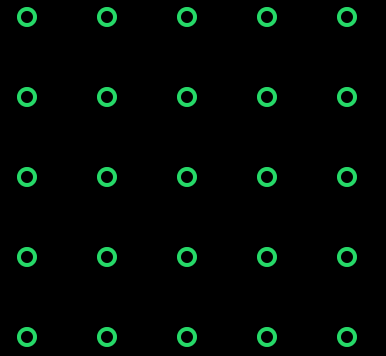


• AI-NATIVE SOFTWARE ENGINEERING



Разработка ПО. *По новым* правилам.

Агентный конвейер берёт рутину. Сеньор-инженеры держат архитектуру.
Первый рабочий прототип от 7 дней.

AI-Native Development & Software Engineering

дек для обсуждения · май 2026

[ПРОДУКТОВЫЙ БРИФ →](#)

Что внутри

01 Где классическая разработка тормозит бизнес

03 Когда разработка становится чёрным ящиком

05 Legacy → современный стек

07 Роли, границы, принципы и упаковка

02 Для чего здесь AI-native подход

04 AI-Native ≠ «GPT» и агентный конвейер

06 Интеграции через MCP

08 Партнёрам и следующий шаг

Где классическая разработка *тормозит* бизнес.

Не каждый проект требует большой команды и месяцев до первой версии. Часто бизнесу нужен рабочий контур быстро — чтобы проверить гипотезу, автоматизировать процесс или заменить устаревший участок системы.

МVP ЖДЁТ 2–4 МЕСЯЦА

Гипотеза устаревает раньше, чем появляется первая рабочая версия.

ВНУТРЕННИЕ ИНСТРУМЕНТЫ ГОДАМИ НЕ ДОХОДЯТ ДО ПРИОРИТЕТА

Команда продолжает жить в Excel, чатах и ручных согласованиях, потому что «основной roadmap важнее».

ИНТЕГРАЦИИ СТАНОВЯТСЯ ОТДЕЛЬНЫМ ПРОЕКТОМ НА МЕСЯЦЫ

CRM, 1С, ERP, база, почта, мессенджеры и API живут отдельно, а люди руками переносят статусы.

LEGACY ДОРОЖАЕТ С КАЖДОЙ НОВОЙ ФИЧЕЙ

Старая система работает, но каждое изменение становится рискованным, долгим и дорогим.

ДОКУМЕНТАЦИЯ И ТЕСТЫ ОТСТАЮТ ОТ КОДА

Проект вроде развивается, но знания остаются в головах, чатах и pull request'ах.

КОМАНДЫ ДЖУНОВ «НЕ ТЯНУТ» СЛОЖНОСТЬ

Архитектурные решения откладываются, технический долг растёт, сроки сдвигаются, ответственность размазывается.

Убираем *лишнюю ручную* работу из конвейера разработки.

AI-native нужен не чтобы заменить инженеров, а чтобы убрать рутину из production pipeline: бриф, спецификация, код, тесты, документация, ревью-черновики и интеграции. Архитектура и ответственность остаются за сеньорами.

ЧТО УСКОРЯЕМ

- Сбор и уточнение требований
- Продуктовый бриф и спецификацию
- Рутинный код по согласованной архитектуре
- Тесты и документацию одновременно с кодом
- Интеграции и boilerplate
- Рефакторинг повторяемых участков
- Подготовку release notes и handover

ЧТО ОСТАЁТСЯ ЗА СЕНЬОРОМ

- Архитектура системы
- Безопасность и комплаенс
- Ключевые технические решения
- Финальное ревью
- Выпуск релиза в продакшен
- Ответственность перед клиентом

Когда разработка становится *чёрным ящиком*.

Для собственника и CEO разработка часто превращается в зону, где всё вроде бы движется, но непонятно: что реально готово, где риски, кто блокирует процесс и почему сроки снова сдвигаются.

ВЫ УСТАЛИ ЛИЧНО КОНТРОЛИРОВАТЬ КОМАНДУ

Каждый статус приходится вытаскивать руками: из чатов, задач, созвонов и личных сообщений.

ВЫ НЕ ВЕРИТЕ ДАШБОРДАМ РАЗРАБОТКИ

Jira / YouTrack / Linear показывают активность, но не отвечают на главный вопрос: «что реально происходит и где риск?».

УШЁЛ СТО ИЛИ TESH LEAD

Вместе с человеком ушёл контекст: архитектурные решения, договорённости, причины компромиссов и часть команды.

ПРОЕКТ ЗАВИСИТ ОТ НЕСКОЛЬКИХ КЛЮЧЕВЫХ ЛЮДЕЙ

Если один человек пропадает, релиз, знания и ответственность начинают рассыпаться.

ДОКУМЕНТАЦИЯ ОТСТАЁТ ОТ РЕАЛЬНОСТИ

Решения живут в чатах и PR, а не в понятной проектной памяти.

РЕШЕНИЕ

AI-native engineering layer: проектная память, weekly engineering brief, контроль рисков, handover и связь между задачами, кодом, документацией и решениями.

AI-Native ≠ «используем *GPT* в разработке».

AI-Native — это когда агентный конвейер встроен в ядро процесса разработки. Код, тесты, документация, рефакторинг — задачи для агента. Архитектура и решения — для инженера.

КЛАССИКА

- Джуниор пишет код, сеньор делает ревью
- QA тестирует вручную, покрытие растёт медленно
- Документация отстаёт от кода
- Сроки линейно зависят от количества людей
- MVP — 2–4 месяца

AI-NATIVE

- Агент пишет рутинный код по спецификации
- Тесты пишутся одновременно с кодом
- Документация — часть генерации
- Скорость растёт от процесса, а не от числа людей
- Первый рабочий прототип от 7 дней

Почему это *не vibe coding*.

Vibe coding помогает быстро собрать черновик. AI-native engineering нужен, когда результат должен жить дальше: развиваться, интегрироваться, проходить ревью, работать с данными клиента и поддерживаться после первого демо.

VIBE CODING

- Один человек быстро собирает демо
- Архитектура рождается «по ходу»
- Код работает «пока его не трогать»
- Тесты и документация — потом, если останется время
- Контекст живёт в голове автора и истории чата с AI
- Сложно передать проект другой команде

DEVNEUROX AI-NATIVE ENGINEERING

- Сначала бриф, score, ограничения и критерии успеха
- Сеньор задаёт архитектуру и границы системы
- Агенты ускоряют рутину: код, тесты, доки, draft-ревью
- Каждый результат проходит инженерный review
- Репозиторий, документация, changelog, handover, support
- Интеграции, роли, доступы и журнал действий — заранее

AI-инструменты стали доступны всем. **Инженерная ответственность — нет.**

Агентный *конвейер*, а не «человек с copilot».

Мы не берём обычную команду и не выдаём ей AI-ассистента. Мы строим конвейер, где агент — исполнитель, а сеньор-инженер управляет и отвечает за результат.

БРИФ

AI-АГЕНТ

Агент интервьюирует команду, уточняет требования, формирует продуктовый бриф.

СПЕКА

АГЕНТ + РЕВЬЮ

Агент пишет спецификацию, сеньор делает ревью.

КОД

АГЕНТ

Агент пишет код, тесты, документацию параллельно.

РЕВЬЮ

СЕНЬОР

Сеньор проверяет архитектуру, безопасность, пограничные случаи.

РЕЛИЗ

АГЕНТ + ЧЕЛОВЕК

Агент готовит релизные шаги и чек-лист деплоя, инженер контролирует выпуск.

От идеи — *до масштаба.*

Одинаковый каркас для MVP и для enterprise. Разница — в глубине каждого этапа и пропорции агент/инженер.

01

Продуктовый бриф

AI-АГЕНТ

Агент интервьюирует команду, собирает требования, формирует продуктовый проспект.

02

AI-Native сборка

РАЗРАБОТКА

Агентный конвейер строит MVP. Большая часть рутинного кода — автоматом. От 7 дней до первого рабочего прототипа.

03

Контроль человека

ИНЖЕНЕРИЯ

Сеньор-инженеры держат архитектуру, ревью, критичные решения, ответственность.

04

Масштаб

ENTERPRISE

Переход к классической инженерии там, где нужна максимальная стабильность и SLA.

AI-Native Development

Первый рабочий прототип от 7 дней. Когда скорость — часть продукта.

Стартапы, продуктовые команды, пилотные проекты. Там, где нужно быстро проверить гипотезу на реальных пользователях и не ждать 4 месяца до первой сборки.

ЧТО ВХОДИТ

- Полный стек: фронтенд, бэкенд, БД
- Интеграции с платежами, CRM, маркетингом
- Деплой в облако (Yandex / VK / Selectel / AWS)
- Документация для команды клиента

ОРИЕНТИРЫ НАПРАВЛЕНИЯ

7 дней

до первого прототипа

2–4 мес

до продакшена

Сеньор

команда

Software Engineering

Там, где скорость и риск — разные вещи

Enterprise и регулируемые отрасли. AI здесь не главный — он ускоряет рутину, но архитектуру, безопасность и compliance держит инженер. Длинный горизонт поддержки.

ЧТО ВХОДИТ

- Архитектура под масштаб и нагрузку
- Бэкенд высокой доступности
- Комплаенс: ФЗ-152, PCI, отраслевые стандарты
- Долгосрочная поддержка со SLA
- Миграция legacy в современный стек

БЭКЕНД • АРХИТЕКТУРА • МАСШТАБ • КОМПЛАЕНС

ОРИЕНТИРЫ НАПРАВЛЕНИЯ

Enterprise

надёжность

SLA

по договору

10+ лет

опыта в стеке

КОМУ: ENTERPRISE, ФИНСЕКТОР, РЕГУЛИРУЕМЫЕ ОТРАСЛИ

Переписываем legacy с *контролем* технического долга.

Команды застревают на старых стеках: каждая новая фича дороже предыдущей, инженеры увольняются, поддержка тяжелеет. Переписать «в лоб» — долго и страшно. Агентный конвейер меняет экономику миграции.

ЧТО УМЕЕМ

- Аудит legacy и дорожная карта миграции
- Параллельная работа: новый код пишется, старый в проде
- Модульная замена без остановки ключевых процессов
- Современный стек и чистая архитектура
- Тест-покрытие агентами одновременно с кодом

РЕЗУЛЬТАТ НА ТИПОВЫХ ПРОЕКТАХ

Быстрее чем переписать «в лоб»

Дешевле в поддержке

Минимум простоев продакшена

Интеграции *быстрее*, чем раньше.

MCP (Model Context Protocol) — открытый стандарт, через который большинство сервисов, CRM, ERP и внутренних баз можно подключить как инструмент агента — через готовый или собственный коннектор.

ГОТОВЫЕ КОННЕКТОРЫ

Slack, GitHub, базы данных, Jira, amoCRM, Bitrix24, Notion, облачные сервисы — подключение за часы.

СВОИ MCP-СЕРВЕРЫ

Под уникальные внутренние системы, 1С, самописные CRM, legacy-API с любым форматом.

ЕДИНЫЙ КОНТРОЛЬ ДОСТУПА

Журнал действий, разграничение прав, шифрование — на уровне протокола, а не в каждой интеграции.

ОБЛАКО И ON-PREM

Работает и с публичными LLM, и с изолированными моделями. Приватный контур — по умолчанию.

AI пишет быстро. *Инженер* отвечает за систему.

Мы не продаём «код от нейросети». Мы строим процесс, где AI ускоряет рутину, а сеньор-инженер отвечает за архитектуру, безопасность, поддержку и результат перед клиентом.

АГЕНТ ДЕЛАЕТ

- Генерация кода по спецификации
- Автотесты параллельно с кодом
- Документация: docstrings, README, описание API
- Рутинный рефакторинг по паттернам
- Первая линия ревью PR
- Тестовые данные и фикстуры

ИНЖЕНЕР ДЕЛАЕТ

- Архитектура системы и модульная разбивка
- Безопасность и комплаенс
- Ключевые технические решения
- Финальное ревью кода перед слиянием
- Ответственность перед клиентом
- Коммуникация с командой клиента

Ясность в том, что *не берём*.

Не берёмся за проекты, где точно не дадим результата или где результат не соответствует нашим принципам.

ЖЕЛЕЗО И ПРОШИВКИ

Мы делаем чистый софт. Прошивки, драйверы, низкоуровневый код — не наш профиль.

АЗАРТНЫЕ ИГРЫ, ADULT, ТЁМНЫЕ ПАТТЕРНЫ

Продукты с этическими проблемами или вводящие пользователя в заблуждение — не наш рынок.

ДИЗАЙН БЕЗ ТЗ

Работаем с готовым дизайн-файлом (Figma). Не делаем одновременно и дизайн, и разработку без брифа.

LEGACY БЕЗ ПЛАНА

Долгосрочная поддержка старого кода без дорожной карты на рефакторинг — сольёте бюджет.

УСТАРЕВШИЕ СТЕКИ

Проекты на стеках без планов миграции — жёсткая архитектурная ловушка на 3–5 лет вперёд.

ПЕРСОНАЛЬНЫЕ ДАННЫЕ НА ПУБЛИЧНЫХ LLM

Работа с персональными данными через публичные облачные модели без согласованного контура и NDA.

От *MVP* до платформы.

Не уточнённая продуктовая ниша — не помеха. Главное — задача и готовность участвовать в брифе.

01

SaaS-продукты

B2B / B2C SaaS, подписка, масштаб

02

Маркетплейсы

Двусторонние платформы, подбор, платежи

03

Внутренние инструменты

Админки, дашборды, корпоративные CRM

04

Интеграции и шины данных

API, очереди, ETL, обмен сообщениями

05

Бэкенд для мобильных

Авторизация, push, офлайн-синхронизация, API

06

Данные и аналитика

Платформы, пайплайны, ML-сервисы

Пять правил, которые *определяют* проект.

Те же правила, что и во всей студии. Здесь — в разрезе процесса разработки.

Релиз > слайды

Результат проекта — работающий код и запущенный продукт, а не презентация.

Среда агентов

Агент встроен в процесс как среда разработки, а не как плагин к IDE.

Сначала приватность

Код, данные и интеллектуальная собственность клиента не покидают согласованный контур.

Честная скорость

Скорость — от отсутствия лишних звеньев, а не за счёт качества.

Решение за человеком

Сеньор принимает архитектурные решения и отвечает за результат.

Четыре формата — *ваш* этап.

Стоимость и SLA — в отдельном КП под конкретную задачу. Здесь — объём и ориентиры.

БРИФ

План и архитектура

- 3–5 дней с командой
- Продуктовый проспект и объём
- Архитектурная схема
- Оценка пилота / MVP

КОМУ

Идея есть, не ясно как и за что браться

MVP-СПРИНТ

Прототип за 7 дней

- Рабочая версия для проверки гипотезы
- Базовый стек + 1–2 интеграции
- Сроки: 1–4 недели
- Поддержка 30 дней

КОМУ

Стартапы, пилотные проекты, проверка гипотезы

ПРОДАКШЕН

Полный цикл до релиза

- Готовый к продакшену SaaS / продукт
- Полный стек + бизнес-интеграции
- Сроки: 2–4 месяца
- Поддержка 90 дней, метрики

КОМУ

Продуктовые команды, B2B-инструменты

ENTERPRISE

Software Engineering и масштаб

- Система под нагрузку и рост
- Комплаенс, ФЗ-152, SLA по договору
- Сроки: от 3 месяцев
- Долгосрочная поддержка, миграция legacy

КОМУ

Enterprise, финсектор, регуляторы

Как партнёр может *продать* AI-native разработку.

Вы приносите клиентский контекст, отрасль и доверие. DevNeuroX берёт AI-native engineering: discovery, архитектуру, пилот, интеграции, поставку и поддержку.

01 • НАЙТИ ПРОЦЕСС

Где у клиента тормозит классическая разработка: MVP, внутренние инструменты, интеграции, legacy, документация.

02 • ПРОВЕРИТЬ FIT

Есть ли владелец процесса, доступ к данным, повторяемость, понятный результат и бюджет на пилот.

03 • УПАКОВАТЬ ПИЛОТ

Формируем объём, сроки, критерии успеха, ограничения, роли клиента и DevNeuroX.

04 • ДОВЕСТИ ДО РЕЗУЛЬТАТА

Запускаем рабочий контур, собираем обратную связь, дорабатываем и решаем: масштабировать или остановить.

WHITE-LABEL DELIVERY • СОВМЕСТНАЯ ПРОДАЖА • ТЕХНИЧЕСКИЙ ПРЕСЕЙЛ • ПИЛОТ ПОД КЛИЕНТА • LONG-TERM ПОДДЕРЖКА

Кейсы — *по запросу.*

Фактуру по конкретным проектам передаём под NDA. Что можем показать и в каком формате — обсудим с учётом вашей отрасли и задачи.

ЧТО ПОКАЖЕМ

- Типы задач, похожие на вашу
- Архитектурные решения (обезличенные)
- стек и ориентир по срокам
- Параметры до / после внедрения

ЧТО НУЖНО ОТ ВАС

- Описание задачи и текущей ситуации
- Контекст отрасли и ограничений
- Приоритет — куда смотрим в первую очередь
- NDA — готовы работать по вашему шаблону

Как мы работаем

| | |
|--------------|--|
| ФОРМАТ | Фиксированный объём + предсказуемая цена по каждому этапу |
| СКОРОСТЬ | Прототип от 7 дней · продакшен 2–4 мес · Enterprise — по плану |
| ПРОЗРАЧНОСТЬ | Регулярные апдейты в Telegram, демо-сессии по этапам |
| ПРИВАТНОСТЬ | Приватный контур: код, данные и интеллектуальная собственность клиента изолированы |
| КОНТРАКТ | По шаблону клиента или DevNeuroX, NDA по умолчанию |
| ОПЛАТА | По этапам. Первый транш — после продуктового брифа |
| ПОДДЕРЖКА | 30–90 дней после релиза включены в стоимость |

Давайте строить *вместе.*

Расскажите о задаче — пришлём Продуктовый проспект и оценку в течение 2 часов.

САЙТ

devneurox.tech

TELEGRAM

[@dxaiblog](https://t.me/dxaiblog)

EMAIL

gk@devneurox.tech

ФОРМА

devneurox.tech/#contact

[ПОГОВОРИ С АГЕНТОМ →](#)