UNDER THE HOOD · ВЫПУСК 01

Code w/ Claude London 2026

24 сессии Anthropic в карте для русскоязычного предпринимателя

ВЫПУСК

FULL для Under the Hood

ДАТА

25 мая 2026

АВТОР

Илья Утов

СОДЕРЖАНИЕ

Восемь разделов. Одна карта дня.

01

ЧТО ПРОИЗОШЛО 19 МАЯ 2026

Введение

CODE W/ CLAUDE LONDON · 19 МАЯ 2026

Один день.
Четыре трека.
Двадцать четыре сессии.

ПРОГРАММА

24

сессии за день

с 9:00 до 17:35

ДЛЯ СМБ

13

из 24 must-watch

54% программы для нетехнической аудитории

МЕТОДОЛОГИЯ

6

runtime patterns

peer-reviewed arXiv 2605.20173

КЕЙСЫ

5

компаний с цифрами

monday, Doctolib, Delivery Hero +

ПРАКТИКА

8

шагов на эту неделю

от Routines до Thinking Lever

ГЛАВНЫЙ СДВИГ ДНЯ

Anthropic за неделю закрыл enterprise-периметр.

«Мы не отдадим данные в облако» больше не аргумент.

ОКНО ОПЕРЕЖЕНИЯ

12 – 18 месяцев

прежде чем новые правила игры станут стандартом российского сегмента

ИСТОЧНИК

claude.com/code-with-claude/london

Verified WebFetch 25.05.2026 · Полная программа из 24 сессий

Анализ и адаптация для русскоязычного СМБ · Under the Hood

AF CLUB

Что произошло 19 мая 2026 года в Лондоне

Anthropic провёл свою вторую конференцию для разработчиков Code w/ Claude. Первая прошла в Сан-Франциско 6 мая, третья запланирована в Токио 5-6 июня. Лондонская версия 19 мая собрала 24 сессии в одном дне, в четырёх параллельных треках, с участием 30 с лишним спикеров от Anthropic и от крупных пользователей платформы.

Конференция прошла в формате полного рабочего дня: с keynote в 9 утра до завершающей сессии в 17:35. Между ними кардинальные анонсы коммерческой стратегии Anthropic, фундаментальные академические доклады о методологии AI-агентов, и серия кейсов от компаний, которые встроили Claude в свою операционку.

Все материалы конференции выложены публично на claude.com/code-with-claude/london. Записи каждой сессии планируются к публикации в течение 2-3 недель после события.

Почему этот PDF существует

В русскоязычном сегменте AI-сообщества о конференции написали два типа публикаций.

Первый тип: новостные заметки с пересказом главных анонсов Anthropic. Это уровень «вот что важно, читайте сами».

Второй тип: списки ссылок на отдельные сессии без анализа. Это уровень «вот пять интересных, остальные посмотрите если будет время».

Обоих типов недостаточно для предпринимателя, который должен принимать решения о применении AI в своём бизнесе на горизонте 5-7 лет. Новостная заметка даёт информацию, но не даёт оптики. Список ссылок даёт ссылки, но не даёт картины целого.

Этот PDF предлагает третий уровень. Полная карта 24 сессий, выжимка стратегического направления Anthropic за этими сессиями, и набор конкретных практик, которые можно применить в течение этой недели.

PDF написан для двух аудиторий. Полная версия 40-50 страниц это материал для Under the Hood, для тех, кто уже работает с AI и принимает архитектурные решения. Lite-версия 15-20 страниц это материал для широкого круга владельцев малого и сред-

него бизнеса, которые слышали об AI и хотят понять с чего начать. Различия между версиями: глубина анализа, объём стратегических разделов, наличие связок с академическими источниками и с нашими внутренними методологическими наработками.

Главный сюжет дня

Если выделить из 24 сессий один сквозной сюжет, он звучит так: Anthropic в Лондоне формализовал свой ход на enterprise-рынок.

До 19 мая Anthropic был «лучшей моделью для разработчиков». Это позиционирование сложилось ещё в 2024 году, и оно работало для аудитории прогрессивных tech-команд, которые сами выбирают инструменты и сами их интегрируют.

19 мая Anthropic явно переключился на другую целевую аудиторию. Не разработчик, а **покупатель с бюджетом в регулируемой отрасли**. CIO банка, CFO страховой компании, COO медицинской платформы, генеральный директор юридической фирмы. Для этой аудитории «лучшая модель» это вторичный фактор. Первичный фактор это совместимость с регуляторными требованиями, ИБ-периметром, корпоративным комплаенсом.

И ровно под эту аудиторию Anthropic выкатил пакет архитектурных функций: Self-Hosted Sandboxes для исполнения задач на инфраструктуре клиента, MCP Tunnels для подключения к внутренним системам без открытия портов наружу, Claude Code Routines для регулярных автоматизированных процессов, 20 с лишним legal MCPs для юридической вертикали, +50% к лимитам Claude Code как прямой ответ OpenAI Codex on-prem.

Сюжет вокруг этого пакета развёрнут в Разделе 2. Главный архитектурный вывод: возражение «мы не выпустим данные в облако», которое блокировало большинство пилотов в регулируемых отраслях с 2024 года, теперь имеет формальный архитектурный ответ.

Параллельный сюжет: методологический слой

Помимо коммерческой стратегии в Лондоне развернулась вторая важная линия. Методология AI-агентов как самостоятельная дисциплина.

До мая 2026 разговор о AI-агентах в индустрии шёл на двух уровнях: технический (какая модель сильнее на бенчмарке) и маркетинговый (у нас есть AI-функция). Третьего уровня, методологического, формально не существовало. Каждый продукт собирали по интуиции архитекторов, без общего языка.

За четыре недели до Лондонской конференции картина начала меняться. 21 марта вышел arXiv-paper 2605.10787, который дал имя главному провалу AI-пилотов: Clean-Slate Bias. Спустя четыре недели вышел arXiv 2605.20173 с каталогом шести runtime patterns. На самой конференции Александр Брикен в докладе The Thinking Lever и Sid Bidasaria в воркшопе Stop Babysitting Your Agents довели методологический разговор до практического уровня: что выбирать, под какую задачу, с каким уровнем автономии.

Это не отдельные академические упражнения. Это формирующийся язык, на котором индустрия будет говорить о AI-агентах в ближайшие два-три года, как программисты в 1990-х перешли на язык design patterns. Методологический слой развёрнут в Разделах 3 и 6.

Что в PDF и как его читать

PDF состоит из 8 разделов.

Раздел 1, Введение. Этот раздел.

Раздел 2, Закрытие enterprise-периметра. Главная коммерческая новость недели. Self-Hosted Sandboxes, MCP Tunnels, Routines, legal MCPs. Что это меняет для российских банков, страховых, медицины, юридических компаний.

Раздел 3, Карта шести runtime patterns. Центральный методологический разворот. Три семейства AI-агентов, шесть паттернов внутри них, чеклист «какой паттерн у вашего агента».

Раздел 4, Пять кейсов с цифрами. Реальные кейсы из сессий конференции. Гипергрост 1→80, monday.com, Doctolib, Delivery Hero, Spotify, Lovable, Legora. Что эти компании сделали и что повторимо для российского СМБ.

Раздел 5, Восемь практик на эту неделю. Конкретные actionable шаги. Routines за 10 минут, MCP Tunnel за 30 минут, картирование внутренней памяти, выбор runtime pattern, прокачка промптов, выбор модели под задачу, дизайн от prompt до production, калибровка распределения мышления.

Раздел 6, Stop babysitting your agents. Контр-нарратив автономии. Три уровня риска и три режима автономии. Связка с шестью паттернами как осознанный инженерный выбор.

Раздел 7, Что Anthropic строит на 18 месяцев. Стратегический финал. Vertical-as-stack против OpenAI universal model. Где у Anthropic скрытое конкурентное преимущество, и где у российского предпринимателя точка вхождения. Только в полной версии PDF.

Раздел 8, Приложение. Полная карта 24 сессий с оценкой релевантности для русскоязычного СМБ. Ссылки на каждую сессию. Группировка по трекам.

Кто стоит за этим PDF

PDF подготовлен в рамках канала Under the Hood. Автор: Илья Утов, AI-практик. 13 лет в IT-бизнесе, Technical Cofounder, ex CTO Senses Media, Asigna. Методологический фундамент PDF: год практики работы с AI-агентами в режиме когнитивного симбиоза, накопленные принципы из 86 кристаллизованных инсайтов, тестирование архитектурных гипотез на реальных бизнес-задачах.

Главный отличительный знак этого PDF от любых других обзоров конференции: тексты разделов написаны на основе как самих сессий, так и параллельной работы автора с теми же темами (память агента, runtime patterns, методология автономии). Это не пересказ, это синтез внешнего материала с внутренним опытом.

Дисклеймер

PDF опубликован 25 мая 2026 года, до публикации записей сессий Anthropic. Часть конкретных цифр и цитат из докладов будет дополнена после публикации записей, ожидаемой в начале июня 2026. Обновлённая версия PDF будет доступна по той же ссылке.

Pre-publish gate автор соблюдал: каждое утверждение о цифрах подтверждено источником, помеченным в разделе. Где данные ещё не опубликованы, это явно указано в тексте.

Если в процессе чтения у вас возник вопрос, поправка, или предложение, напишите автору в Telegram-канал @gorilla_under_hood. Каждая обратная связь учитывается в следующих итерациях.

ХРОНОЛОГИЯ ДНЯ

19 мая 2026. Что и когда говорил Anthropic.

09:00 — Main stage

1. **Opening Keynote**
   Ami Vora, Dianne Penn, Angela Jiang, Katelyn Lesse, Cat Wu, Boris Cherny

10:30 — Утренние сессии

2. CLAUDE PLATFORM
   Memory and Dreaming
   Ravi Trivedi · self-learning agents

   RESEARCH WORKSHOP
   Picking the Right Model
   Lucas Smedley · hands-on техники

11:15

3. RESEARCH · MAIN
   Coding не constraint
   Niklas Gustavsson · Spotify devex

   RESEARCH
   Designing with Claude
   Dan Cary · prompt to production

12:00 — Перед обедом

4. CLAUDE CODE · MAIN
   Production faster
   M. Cohen + H. Stall · Managed Agents

   CLAUDE CODE
   From 1 to 80
   Grinberg + Orlev · hypergrowth

12:30 — Workshop

5. WORKSHOP · CLAUDE CODE
   Stop babysitting your agents · Sid Bidasaria

13:50

6. CLAUDE PLATFORM
   How Lovable vibecodes
   Fabian Hedin · production scale

   RESEARCH
   The Thinking Lever
   Alexander Bricken · рычаг мышления

14:35 — Кейсы

7. CLAUDE PLATFORM · MAIN STAGE
   monday.com + Doctolib + Delivery Hero
   Semenov, Kaluzny, Schäfer · три отрасли один паттерн

15:20

8. RESEARCH
   The Capability Curve
   Jeremy Hadfield · кривая возможностей

   CLAUDE CODE
   Signals that trade themselves
   Tushara Fernando · regulated firm

16:05

9. CLAUDE PLATFORM · MAIN
   What legal agents inherit from coding agents
   Jakob Emmerling · Legora

16:50 — Финал дня

10. RESEARCH · ЗАКРЫТИЕ
    The Prompting Playbook · Margot van Laar

17:35 — Evening reception

END

8 часов чистого контента.
24 разговора, которые изменят 2026 год.

Полная программа: claude.com/code-with-claude/london · Under the Hood

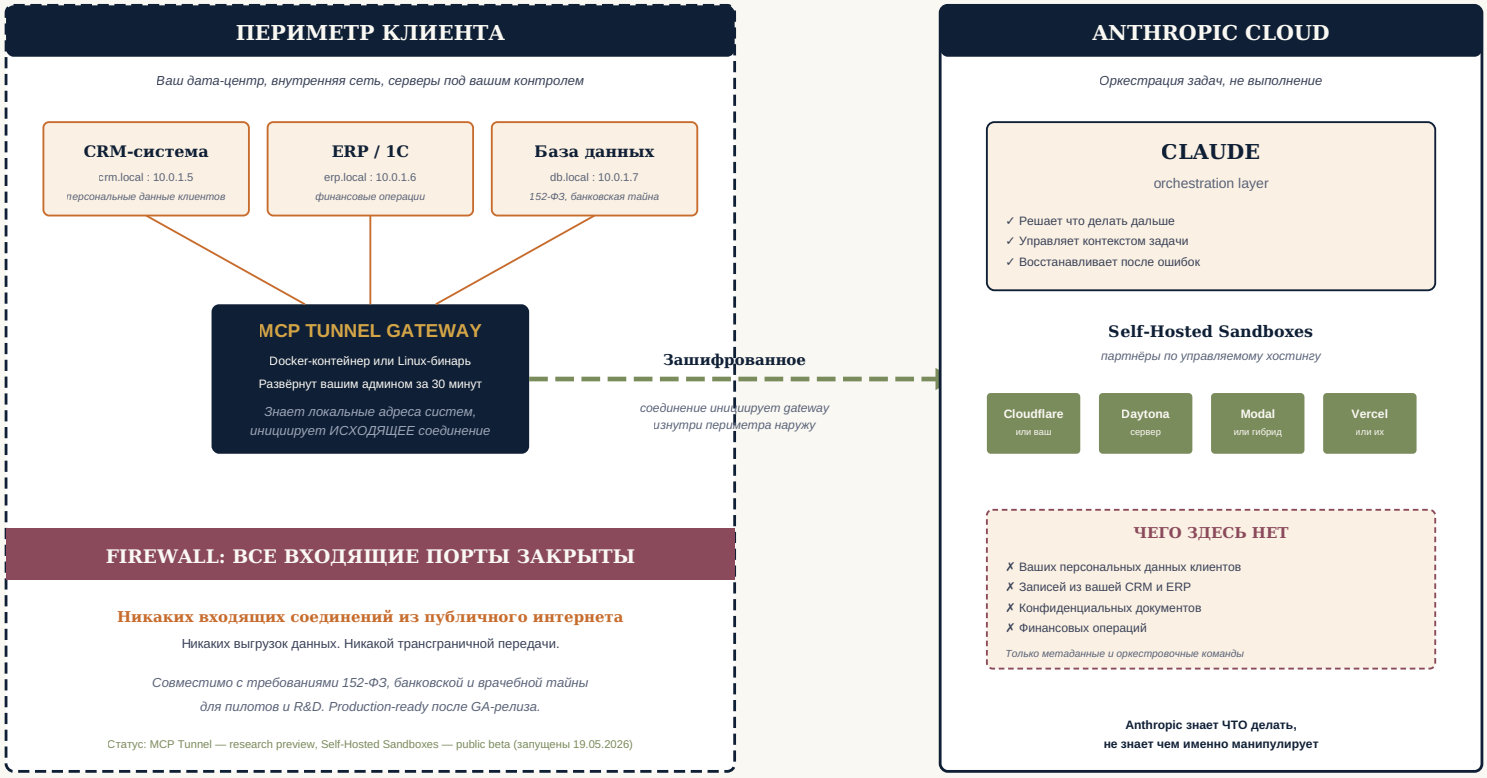
02

ГЛАВНАЯ КОММЕРЧЕСКАЯ НОВОСТЬ НЕДЕЛИ

Закрытие enterprise-периметра

Архитектура MCP Tunnel и Self-Hosted Sandboxes

Как Claude работает с внутренними системами клиента без выноса данных в облако



Источник: Anthropic Help Center, Code w/ Claude London 19.05.2026 (Michael Cohen, Harrison Stall) — Under the Hood

Запущены 19 мая 2026. MCP Tunnels: research preview. Self-Hosted Sandboxes: public beta.

Возражение, которое держало рынок

Возьмём типичный разговор владельца российского бизнеса со службой безопасности по поводу AI.

Владелец говорит: «Давайте попробуем Claude для отдела поддержки. Время ответа клиенту упадёт с двух часов до двух минут. Команда освободится для сложных кейсов».

Служба безопасности отвечает: «Где будут жить данные клиентов в момент обработки?»

Владелец: «В облаке Anthropic в США».

Служба безопасности: «У нас 152-ФЗ. Персональные данные граждан России обрабатывать в облаке за рубежом мы не имеем права без согласия Роскомнадзора и формальной процедуры трансграничной передачи. Это полгода работы юристов, и в конце Роскомнадзор всё равно может сказать нет».

Владелец: «Тогда пилот в банковской системе? У нас банковская тайна».

Служба безопасности: «Банковская тайна тем более. ЦБ не пропустит обработку в зарубежном облаке».

Владелец: «Тогда что?»

Служба безопасности: «Тогда никак. AI можно после того, как появится способ обрабатывать данные внутри нашего периметра».

Этот разговор повторялся в каждой второй компании с 2024 по апрель 2026 года. Не цена была блокером. Не качество модели. Не количество интеграций. Именно архитектурная проблема: где физически живут данные клиента в момент обработки.

Anthropic 19 мая 2026 года эту проблему решил архитектурно. Не «у нас SOC2-сертификат», не «мы шифруем данные при передаче». А «данные могут вообще не покидать ваш периметр».

Что выкатил Anthropic за пять дней

С 19 по 23 мая Anthropic выпустил пять связанных функций, которые в сумме закрывают enterprise-возражение. Разберём каждую.

Self-Hosted Sandboxes (публичная бета)

Sandboxes это среда исполнения задач AI-агента. До 19 мая Anthropic управлял этой средой целиком: оркестрация задач, исполнение шагов, восстановление после ошибок происходили на серверах Anthropic в США.

С 19 мая среда исполнения разделена. Оркестрация остаётся у Anthropic. Исполнение переносится на инфраструктуру клиента. У клиента есть три варианта.

Первый: партнёры по управляемому хостингу. Cloudflare, Daytona, Modal, Vercel. Это означает, что вы выбираете одного из этих провайдеров (любой подходит под комплаенс ЕС или США в зависимости от вашего региона), даёте ему ваш API-ключ Anthropic, и он автоматически разворачивает sandbox внутри своей инфраструктуры под вашим аккаунтом. Ваши данные физически живут в инфраструктуре провайдера, не в облаке Anthropic.

Второй: ваш собственный сервер. Anthropic выдаёт Docker-образ sandbox. Вы разворачиваете его внутри своей сети, на вашем железе. Данные не покидают периметра компании ни на секунду.

Третий: гибрид. Часть задач исполняется в облаке Anthropic (для скорости и нетребовательных к данным сценариев), часть на вашей инфраструктуре (для чувствительных). Один и тот же AI-агент работает в двух режимах в зависимости от метки задачи.

Self-Hosted Sandboxes находятся в публичной бете. Это не production-ready GA, но для пилотов и для построения архитектурного proof-of-concept это уже работает.

MCP Tunnels (research preview)

MCP это Model Context Protocol, стандарт, по которому AI-агент Anthropic подключается к вашим внутренним системам: CRM, ERP, базы данных, файловое хранилище. До 19 мая, чтобы подключить агента к внутренней системе, нужно было открыть порт на firewall и впустить входящий трафик из публичного интернета внутрь компании. Любой ИБ-офицер от такого должен возражать.

MCP Tunnels работают иначе. Это исходящее зашифрованное соединение. Внутри вашего периметра ставится маленький шлюз. Шлюз сам инициирует исходящее соединение к Anthropic. Через это соединение и идёт коммуникация. Никаких входящих подключений из публичного интернета. Никаких открытых портов на firewall. Никаких выгрузок данных. Ваши CRM, ERP и базы данных остаются внутри периметра.

Архитектурная аналогия: это как звонок изнутри здания наружу через защищённую IP-телефонию, в отличие от прежней модели, где нужно было пускать входящие звонки на внутренний номер. Кто инициирует соединение, тот контролирует контекст.

MCP Tunnels находятся в research preview. Это ранее, чем публичная бета. Для пилотов работает, для production-критичной нагрузки рекомендуется дождаться GA, который ожидается осенью 2026.

▸ Claude Code Routines

Routines это запланированные задачи. То, что в IT-инфраструктуре делается через cron, scheduled tasks, или специализированные workflow-системы типа Airflow. У Anthropic теперь это встроено в Claude Code.

Лимиты по подпискам: - Pro: до 5 routines в день - Max: до 15 routines в день - Team и Enterprise: до 25 routines в день

Что это даёт владельцу бизнеса без технической команды. Любую повторяющуюся работу, на которую раньше нужен был разработчик и сервер, теперь можно настроить из веб-интерфейса за 10-15 минут. Каждое утро Claude собирает упоминания бренда за сутки и кладёт в почту. Каждый понедельник сверяет остатки в 1С с продажами на маркетплейсе и присылает несовпадения в Telegram. Каждый вечер делает короткое summary за день из писем ключевых клиентов.

Routines работают в production. Это не бета.

▸ +50% к лимитам Claude Code

С 13 мая 2026 года и до 13 июля 2026 года Anthropic увеличил лимиты использования Claude Code на 50% для всех подписок: Pro, Max, Team, Enterprise. Это формально «временная мера», но стратегически это прямой ответ на запуск OpenAI Codex on-prem через Dell 18 мая. Anthropic поднимает порог потребления, чтобы переключение на конкурента стало менее привлекательным экономически.

С точки зрения покупателя: если вы планируете внедрить Claude Code в команде, окно до 13 июля даёт +50% к лимитам бесплатно. Это окно для расширения использования.

▸ Legal MCPs (20+ коннекторов и 12 plugins)

С 20 мая Anthropic запустил 20 с лишним legal MCP-коннекторов и 12 plugins для юридических практик. Это интеграции под популярные международные юридические системы: LexisNexis, Westlaw, Clio, NetDocuments и подобные. Для российского рынка прямой релевантности меньше, но архитектурный сигнал важный.

На той же неделе, по данным самого Anthropic, legal стал функцией Claude Cowork номер один по engagement среди всех профессиональных доменов. То есть legal-юзеры пользуются Cowork активнее, чем разработчики, маркетологи, операционисты. Связан с этим кейс Legora (см. раздел «Кейсы»), показавший как legal-агенты унаследовали архитектуру от code-агентов.

Почему именно сейчас

Это не случайный апдейт. 18 мая OpenAI вместе с Dell представили Codex on-prem hybrid: вычисления в облаке OpenAI плюс выполнение задач на серверах Dell внутри периметра клиента. Это был первый удар по сегменту регулируемых отраслей со стороны OpenAI.

Anthropic ответил за неделю. Не «у нас лучше модель», не «у нас дешевле тариф». Архитектурно: ваш периметр включён в наш стек как первоклассный гражданин. И не партнёрство с одним поставщиком (как Dell у OpenAI), а открытая архитектура с четырьмя партнёрами по выбору клиента.

Стратегически это окончание игры «у кого модель умнее». 2026 год это игра в «у кого инфраструктура совместима с регулируемыми отраслями». Anthropic закрыл главную линию обороны.

Что это меняет для российского бизнеса

Конкретные следствия для трёх сегментов.

Для банков и страховых. Раньше путь к Claude закрывался первым же вопросом ИБ. Теперь есть формальный архитектурный ответ. Self-Hosted Sandbox даёт обработку на вашей стороне. MCP Tunnel даёт связь с внутренними системами без открытия портов наружу. Это не означает, что комплаенс автоматически пройдён. Но это означает, что есть, что показать комитету: чёткая архитектура, не маркетинговая декларация.

Для медицины и всего под 152-ФЗ. Персональные данные могут физически не покидать ваш периметр. Это снимает основной риск трансграничной передачи. Аудит протоколов внутри вашей сети возможен. Чувствительные процессы, такие как обработка обращений пациентов, медицинская документация, recommendation systems для врачей, можно запускать в пилоте уже сейчас, в режиме research preview.

Для юристов в малом и среднем бизнесе. 20+ legal MCP-коннекторов это готовые интеграции под популярные международные юридические системы. Прямое применение для российской практики ограничено разницей в правовых системах, но техниче-

ская архитектура переносима один к одному. Если вы планируете кастомные интеграции с базой судебной практики, базой контрагентов, базой типовых документов, у вас теперь есть архитектурный шаблон.

Что делать в течение недели

Шаг первый. Откройте документацию Anthropic по MCP Tunnels и Self-Hosted Sandboxes. Это 30 минут чтения. После этого у вас будет язык для разговора со службой безопасности.

Шаг второй. Если у вас уже работает AI на одном поставщике, оцените стоимость и сроки multi-vendor стратегии. На горизонте 12-18 месяцев это станет нормой для регулируемых отраслей, и тот, кто внедрит первым, получит конкурентное преимущество в найме, продажах, и переговорах с регулятором.

Шаг третий. Если вы только выбираете, запустите пилот в одном изолированном процессе с MCP Tunnel. Простой кейс: внутренний справочник по продуктам, помощник для отдела поддержки, обработка типовых запросов клиентов. Через две недели у вас будет реальный опыт работы новой архитектуры в вашей инфраструктуре, не теоретическая оценка.

Через год это будет стандартом российского сегмента. Сейчас это окно для того, кто внедрит первым.

03

PEER-REVIEWED МЕТОДОЛОГИЯ

Карта шести runtime patterns

Шесть runtime patterns для AI-агентов в проде

arXiv 2605.20173 + The Thinking Lever (Code w/ Claude London, 19.05.2026)

Семейство 1. Разговорные

Шаг = команда пользователя + ответ агента

1 Чистый chat

Модель отвечает свободным текстом.
Нет вызова инструментов.

Примеры: базовый ChatGPT, Claude без MCP, simple support-боты Intercom и Zendesk

УРОВЕНЬ РИСКА: НИЗКИЙ

2 Tool-augmented chat

Модель может вызвать инструмент:
поиск, код, базу, API. Встраивает результат.

Примеры: ChatGPT с tools, Claude с MCP, Perplexity, Cursor custom agents

УРОВЕНЬ РИСКА: НИЗКИЙ-СРЕДНИЙ

Семейство 2. Автономные

Получил цель — идёт к результату сам

3 ReAct-агент

Чередует размышление — действие — наблюдение, до результата.

Примеры: AutoGPT, Manus, custom-агенты на LangChain и AutoGen

УРОВЕНЬ РИСКА: СРЕДНИЙ

4 Plan-and-execute

Сначала строит план из шагов,
потом исполняет. При сбое перепланирует.

Примеры: BabyAGI, OpenAI Swarm, Claude Code в автономном режиме

УРОВЕНЬ РИСКА: СРЕДНИЙ-ВЫСОКИЙ

Семейство 3. Долгий горизонт

Недели и месяцы, не одна сессия

5 Stateful agent

Состояние хранится во внешней памяти,
подбирается при старте сессии.

Примеры: Devin, Replit agents, CLAUDE.md + memory/ (наша система)

УРОВЕНЬ РИСКА: СРЕДНИЙ

6 Multi-agent + координатор

Несколько специализированных агентов
под управлением orchestrator-агента.

Примеры: Anthropic research-агенты, AutoGen multi-agent, Crewbase

УРОВЕНЬ РИСКА: ВЫСОКИЙ

Граница «модель → детерминированный код»

1 2 3 4 5 6

Граница ШИРОКАЯ
Модель решает всё, что говорить.
Код решает только когда вызывать инструмент.
Гибкость высокая, предсказуемость низкая

Граница СРЕДНЯЯ
Модель строит план или принимает решения.
Код решает порядок и обработку ошибок.
40-60% детерминированного кода

Граница УЗКАЯ
Код управляет состоянием и координацией.
Модель используется точечно.
Стабильность высокая, гибкость низкая

Источник: arXiv 2605.20173 + Code w/ Claude London 19.05.2026 (Alexander Bricken, The Thinking Lever)

Адаптация для русскоязычного СМБ — Under the Hood

Почему именно сейчас появилась карта

До мая 2026 разговор о AI-агентах в проде шёл на двух уровнях. Технический: «какая модель сильнее на бенчмарке». Маркетинговый: «у нас есть AI-функция». Третьего уровня, методологического, в индустрии формально не существовало. Каждый продукт собирали по интуиции архитекторов, без общего языка.

За четыре недели картина изменилась. Сначала, 21 марта 2026, вышел arXiv-paper 2605.10787, который дал имя главному провалу AI-пилотов. Авторы назвали его Clean-Slate Bias: агент считает рабочую среду пустой по умолчанию и удаляет данные клиента. В исследовании было показано, что именно этот провал объясняет до 60% случаев, когда AI-проект «работал в демо и сломался в проде». Это был первый peer-reviewed термин для практической проблемы, которую до этого описывали туманно как «нестабильность».

19 мая на конференции Code w/ Claude London Александр Брикен из Anthropic выступил с докладом The Thinking Lever. Он говорил о том, какую часть работы вы оставляете себе, а какую отдаёте Claude, и где между ними рычаг. Спустя несколько дней вышел второй академический документ, arXiv 2605.20173, под названием «A Methodology for Selecting and Composing Runtime Architecture Patterns for Production LLM Agents». Авторы дали каталог из шести runtime patterns, по которым делятся все AI-агенты, существующие в проде.

Два академических документа за четыре недели на одну тему. Тема методологии. Это первый этап S-кривой: легитимизация языка через peer-review. На следующем этапе индустрия начнёт говорить на этом языке как на стандартном. Окно для того, кто встроит этот язык в свой продукт раньше других, открыто прямо сейчас.

В программной инженерии это уже было. В 1994 году вышла книга «Design Patterns» (Gamma, Helm, Johnson, Vlissides), которая ввела 23 паттерна объектно-ориентированного дизайна. До этого каждая команда говорила о коде на своём языке, после, индустрия за два года перешла на общий словарь. Шесть runtime patterns для AI-агентов идут той же дорогой.

Три семейства, шесть паттернов

Все AI-агенты делятся на три семейства по способу взаимодействия с задачей.

▸ Семейство первое: разговорные агенты

Агент работает в диалоге. На каждом шаге пользователь даёт команду, агент отвечает. Сюда попадают паттерны 1 и 2.

Паттерн 1: чистый chat-agent. Модель отвечает свободным текстом. У неё нет доступа к инструментам, поиску, базам данных. Это самая простая архитектура, и для большинства практических задач её достаточно. Примеры в проде: ChatGPT в режиме без plugins, Claude в режиме без MCP, базовые support-боты в Intercom или Zendesk.

Паттерн 2: tool-augmented chat-agent. Модель в ходе ответа может вызвать инструмент: поиск по интернету, выполнение кода, обращение к базе, отправка запроса в API. Полученный результат встраивается в ответ. Это самая распространённая архитектура в продуктах 2026 года. Примеры: ChatGPT с включёнными tools, Claude с MCP-серверами, Perplexity, кастомные agents в Cursor.

Граница между моделью и кодом в этом семействе широкая. Модель решает, что говорить и какой инструмент вызвать. Код решает только, как именно выполнить вызов.

▸ Семейство второе: автономные агенты

Агент получает цель, дальше идёт к ней без участия пользователя до результата или ошибки. Сюда попадают паттерны 3 и 4.

Паттерн 3: ReAct-агент. Модель чередует размышление, действие, наблюдение результата, до получения ответа. После каждого шага она оценивает: «продвинулся ли я к цели». Если да, делает следующее действие. Если нет, меняет стратегию. ReAct это аббревиатура от «Reasoning + Acting», предложенная в paper 2210.03629 в 2022 году. Примеры в проде: AutoGPT, Manus, многие custom-агенты на LangChain и AutoGen.

Паттерн 4: plan-and-execute. Модель сначала строит план из шагов, потом исполняет план по очереди. Если на шаге происходит сбой, она перепланирует. На бумаге паттерны 3 и 4 похожи, на практике это совершенно разные звери по поведению в проде. ReAct гибче, но менее предсказуем. Plan-and-execute предсказуемее, но плохо адаптируется к неожиданным результатам. Примеры в проде: BabyAGI ранних версий, OpenAI Swarm (некоторые конфигурации), Claude Code в режиме автономного исполнения сложных задач.

Граница между моделью и кодом в этом семействе уже. Модель строит план или принимает решения, код решает в каком порядке исполнять шаги и что делать при ошибке. Хорошо построенный autonomous-агент имеет 40-60% детерминированного кода, не больше.

▸ Семейство третье: агенты долгого горизонта

Задачи длятся неделями или месяцами, не одной сессией. Сюда попадают паттерны 5 и 6.

Паттерн 5: stateful agent с внешней памятью. Состояние агента сохраняется во внешней системе хранения (база данных, файловая система, vector store) и подбирается при старте новой сессии. Это паттерн, по которому работают long-term assistants в Devin, агенты Replit, агенты для долгих research-задач. Пример из нашего собственного опыта: система памяти в этом проекте (CLAUDE.md + папка memory/) построена ровно по паттерну 5.

Паттерн 6: multi-agent с координатором. Несколько специализированных агентов работают над общей задачей под управлением orchestrator-агента. Координатор раздаёт задачи, отслеживает прогресс, собирает результаты. Это самая сложная архитектура из всех шести, и она оправдана только тогда, когда задача не разрешима на одном агенте даже с долгим горизонтом. Примеры в проде: Anthropic’s research-агенты, AutoGen в multi-agent конфигурации, корпоративные системы класса Crewbase.

Граница между моделью и кодом в этом семействе самая узкая. Код управляет состоянием, передачей контекста, координацией. Модель используется точечно, в местах, где требуется креативное или контекстное решение.

Главный вопрос: где у вашего агента граница

Самый важный архитектурный вопрос 2026 года звучит так: где у вашего AI-агента проходит граница между моделью и детерминированным кодом.

Чем больше решений принимает модель, тем гибче агент. Он адаптируется к неожиданным входным данным, обрабатывает edge cases без переписывания, понимает контекст. Но вместе с гибкостью растёт непредсказуемость. Два запуска агента на одном и том же входе могут дать разные результаты. Тестирование становится статистическим, не детерминированным.

Чем больше кода, тем стабильнее агент. Те же входные данные всегда дают тот же выход. Тестирование классическое. Сбои легко отлаживаются. Но вместе со стабильностью растёт стоимость изменений: любое новое поведение требует написания нового кода.

Александр Брикен в докладе The Thinking Lever сформулировал это как калибровку рычага. На одном конце рычага лежит модель, на другом код. Где именно стоит точка опоры, определяет цену каждой ошибки и стоимость каждого изменения. Для каждой бизнес-задачи точка опоры своя.

Эмпирическое правило, выведенное Бриккеном из практики: чем выше цена ошибки, тем больше детерминированного кода. Чем шире разброс возможных входных данных, тем больше работы модели. Где оба требования высоки одновременно, там вам нужен паттерн 4 (plan-and-execute) или паттерн 6 (multi-agent с координатором). Они дают модели стратегическую гибкость, но удерживают исполнение в строгих рамках кода.

Чеклист «какой паттерн у вашего агента»

Если вы не знаете, какой именно паттерн использует ваш существующий или планируемый AI-агент, прогоните его через четыре вопроса.

Вопрос 1. Агент работает в диалоге или сам идёт к цели без вмешательства? - Диалог: семейство 1 (паттерны 1 или 2). - Сам идёт: семейство 2 или 3.

Вопрос 2. Если сам идёт, помещается ли задача в одну сессию (минуты или часы)? - Помещается: семейство 2 (паттерны 3 или 4). - Не помещается, длится дни, недели, месяцы: семейство 3 (паттерны 5 или 6).

Вопрос 3. Если работает в диалоге, может ли агент вызвать инструмент в процессе ответа (поиск, базу, API)? - Нет, только текст: паттерн 1, чистый chat. - Да: паттерн 2, tool-augmented chat.

Вопрос 4. Если автономный, агент сначала планирует, потом исполняет, или чередует мысль с действием по ходу? - Чередует: ReAct, паттерн 3. - Сначала план, потом исполнение: plan-and-execute, паттерн 4.

Если ваш агент попадает в семейство 3, вопрос 5: один агент с памятью или несколько с координатором? - Один: паттерн 5, stateful. - Несколько: паттерн 6, multi-agent.

Что делать с этим знанием

Перед тем как заказать разработку AI-агента у подрядчика, попросите его в одном предложении назвать паттерн вашего агента. Если подрядчик отвечает «у нас своя архитектура» или «мы делаем умного агента», есть высокий риск, что подрядчик не знает что строит. И тогда у вас будет шанс попасть в те 60% пилотов, которые проваливаются на Clean-Slate Bias или похожих ошибках.

Если у вас уже работает агент, проверьте соответствие паттерна задаче. Разговорный паттерн на задаче «обработай 10 000 заказов за ночь» это пустая трата денег: вам нужен паттерн 4. Автономный паттерн на простом FAQ это ракета по воробьям: вам достаточно паттерна 1 или 2. Multi-agent на задаче из трёх шагов это инженерное самообольщение: один stateful agent справится дешевле и надёжнее.

Каждый паттерн стоит своих денег. Каждый закрывает свой класс задач. Карта существует. Спрашивайте подрядчиков по карте.

Дальнейшее чтение

- arXiv 2605.20173 «A Methodology for Selecting and Composing Runtime Architecture Patterns for Production LLM Agents»
- arXiv 2605.10787 «ComplexMCP: Clean-Slate Bias»
- arXiv 2210.03629 «ReAct: Synergizing Reasoning and Acting in Language Models»
- Запись доклада Alexander Bricken «The Thinking Lever», Code w/ Claude London 19.05.2026
- Gamma, Helm, Johnson, Vlissides, «Design Patterns: Elements of Reusable Object-Oriented Software» (1994)

04

РЕАЛЬНЫЕ ВНЕДРЕНИЯ С КОНФЕРЕНЦИИ

Пять кейсов с цифрами

КЕЙСЫ С КОНФЕРЕНЦИИ

Пять компаний. Один общий паттерн встраивания AI.

КЕЙС 01 · ГИПЕРГРОСТ

От 1 человека до 80

Инжиниринг-команда на Claude Code как основной рабочей среде.
Документация написана не для людей, а для Claude. Code review автоматизирован.

Спикеры: Gabriel Grinberg + Yoav Orlev

1 → 80
ЧЕЛОВЕК — ЛЮДЕЙ

КЕЙСЫ 02-04 · ТРИ ОТРАСЛИ · ОДИН ПАТТЕРН

КЕЙС 02 · SAAS

monday.com

200K+ платящих клиентов

Что сделали:
Claude встроен как функция-помощник внутри workflows клиентов.

Не chatbot сбоку. Часть процесса.

КЕЙС 03 · МЕДИЦИНА

Doctolib

80M+ пользователей в 15 странах ЕС

Что сделали:
Claude обрабатывает первичные обращения пациентов до врача.

Не диагностика. Сортировка по срочности.

КЕЙС 04 · ОПЕРАЦИИ

Delivery Hero

70+ стран, 1.5B заказов/год

Что сделали:
Маршруты, прогноз спроса, обработка нестандартных жалоб.

Не замена операторов. Снижение нагрузки.

КЕЙС 05 · LEGAL

Legora.

Legal-агенты унаследовали архитектуру code-агентов.

Контекст проекта + специализированные инструменты + полный аудиторский след действий.

CLAUDE COWORK
#1
по engagement
среди всех профессий

ОБЩИЙ ПАТТЕРН

AI не заменяет функцию целиком.

AI вставляется в конкретное место workflow.

Где сотрудник тратит максимум времени на повторяемые действия.
И где ошибка имеет ограниченную цену.

Применимо к monday, Doctolib, Delivery Hero, Legora и 1→80.

ФОРМУЛА ДЛЯ ВАШЕГО БИЗНЕСА

Не «давайте сделаем AI-стратегию».
А «давайте найдём 3-5 мест, где можно вставить Claude точечно».

Проверяемо. Дёшево. Повторимо.

Конференция Code w/ Claude London собрала пять компаний из разных отраслей, чтобы каждая рассказала свой опыт встраивания Claude. Это не маркетинговые презентации. Это технические доклады, в которых спикеры показывают конкретную архитектуру решений и конкретные результаты.

Пять кейсов в этом разделе: - Кейс #1. Гипергрост 1→80 на Claude Code (Gabriel Grinberg, Yoav Orlev) - Кейс #2. monday.com (Ruslan Semenov) - Кейс #3. Doctolib (Alex Kaluzny) - Кейс #4. Delivery Hero (Rodrigue Schäfer) - Кейс #5. Legora и legal-агенты (Jakob Emmerling)

Бонусные кейсы: Spotify (Niklas Gustavsson) и Lovable (Fabian Hedin) даны как граничные примеры с собственной структурой.

Кейс #1. От одного человека до 80 на Claude Code

Спикеры: Gabriel Grinberg, Yoav Orlev, со-основатели стартапа [имя компании из записи доклада] **Сессия:** «From one person to 80: Scaling a hypergrowth engineering org with Claude Code» **Track:** Claude Code, Breakout stage

Контекст

В классической модели рост инженерной команды упирается в три вещи: найм, онбординг, потеря качества кода при увеличении численности. Каждая из них стоит реальных денег.

Найм senior разработчика на свободном рынке: 3-6 месяцев поиска, комиссия рекрутёрам 15-25% годовой зарплаты, плюс потерянная выручка из-за того, что задачу никто не делает.

Онбординг до первого продуктивного commit’а: 3 месяца в типичной компании. В стартапе с плохой документацией доходит до 6 месяцев.

Потеря качества при росте: классическая «закон Конвея в действии». Каждые 10 новых разработчиков приносят новые недопонимания архитектуры. К 50 человекам у вас уже не одна кодовая база, а три, и они расходятся.

Grinberg и Orlev на конференции показали другой подход.

▸ Что они сделали

Документация архитектуры, договорённостей команды, типичных ошибок написана не для людей, а для Claude. В корне репозитория файл `CLAUDE.md` с архитектурными решениями. Папка `docs/decisions/` с протоколами прошлых архитектурных выборов. Файлы стандартов кода, рассчитанные на чтение Claude и применение Claude.

Когда приходит новый разработчик, он с первого дня работает в среде, где Claude знает архитектуру не хуже сеньора с двухлетним стажем в компании. Новичок задаёт Claude вопрос «как у нас принято делать X», получает ответ с примером из реальной кодовой базы и со ссылкой на протокол решения. Это снимает 70-80% типичных вопросов, на которые в обычной компании уходят часы старших инженеров.

Code review автоматизирован. Первый проход ревью даёт Claude: проверяет соответствие принятому стилю, типичные баги, покрытие тестами, документацию изменений. Человек подключается только к архитектурным вопросам и финальной подписи. Это снимает с сеньоров рутинную часть ревью и ускоряет цикл от PR до мерджа.

▸ Цифры

[К заполнению из записи доклада: конкретный темп найма, сокращение онбординга, метрики качества кода. Ожидается публикация записи Anthropic в начале июня 2026.]

▸ Что повторимо для российского СМБ

Подход не требует, чтобы у вас уже была инженерная команда. Если у вас сейчас 1-2 разработчика и нужно расти, тот же принцип работает.

Главное: документация архитектуры и договорённостей пишется в формате, который Claude использует с первого запроса. Не отдельный wiki для людей, а файл `CLAUDE.md` в корне репозитория плюс директория `docs/decisions/` с прошлыми решениями.

Применять кейс можно поэтапно. Сначала описать в `CLAUDE.md` существующую архитектуру (1-2 дня работы старшего разработчика). Потом начать вести `docs/decisions/` при каждом архитектурном выборе (по 30 минут на запись). Через 2-3 месяца у нового сотрудника появится среда обучения, которой не было раньше.

Кейс #2. monday.com, AI в core продукт

Спикер: Ruslan Semenov, monday.com **Сессия:** «Building AI-native at enterprise scale» (совместно с Doctolib и Delivery Hero) **Track:** Claude Platform, Main stage

Контекст компании

monday.com это платформа управления работой для команд. По публичным данным на 2025-2026: около 200 000 платящих клиентов, 200+ миллионов пользователей, оценка компании на NYSE более 10 миллиардов долларов.

С точки зрения продукта monday.com конкурирует с Asana, ClickUp, Notion, Jira. Это рынок с высокой конкуренцией по фичам, где разница в качестве продукта определяет рост числа клиентов.

Что они сделали

Semenov показал две архитектурные особенности встраивания Claude.

Первая: Claude встроен не как chat-bot сбоку (как делают многие SaaS-компании в 2025-2026), а как функция-помощник прямо внутри пользовательских workflows. Когда клиент настраивает новую доску задач, Claude предлагает структуру под конкретный тип команды (engineering / sales / marketing / operations). Когда команда долго не закрывает задачу, Claude подсказывает причину блока на основе сообщений в комментариях.

Вторая: Claude используется не только для текстового вывода, но для предсказания и предотвращения проблем в работе команды. Например, если в комментариях к задаче нарастает «температура» (тон становится напряжённым), Claude предлагает руководителю проекта вмешаться раньше, чем конфликт станет открытым.

Цифры

[К заполнению из записи доклада: метрики adoption нового AI-функционала, влияние на retention и upgrade-конверсию.]

Что повторимо для российского СМБ

Для большинства русскоязычных компаний 200 000 клиентов это нереалистичный масштаб. Но архитектурный принцип «AI как функция внутри workflow, не как chat-bot сбоку» переносим один к одному.

Применять можно так. Возьмите ваш самый часто используемый рабочий процесс в компании (CRM-карточка клиента, оформление заказа, согласование документа). Найдите в нём 2-3 точки, где сотрудник тратит время на повторяющиеся текстовые действия (написать описание, составить ответ, выделить важное из истории). Встройте Claude в эти точки как часть процесса, не как отдельную кнопку «спросить AI».

Кейс #3. Doctolib, обработка обращений пациентов

Спикер: Alex Kaluzny, Doctolib **Сессия:** «Building AI-native at enterprise scale» **Track:** Claude Platform, Main stage

▸ Контекст компании

Doctolib это европейская медицинская платформа. Публично на 2025: 80+ миллионов пользователей, 400 000+ медицинских специалистов, работает в 15 странах ЕС (Франция, Германия, Италия, Австрия, Швеция и другие).

Бизнес-модель: SaaS для врачей и клиник плюс маркетплейс пациент-врач с записью на приём. Регуляторный контекст жёсткий: персональные медицинские данные, EU AI Act, GDPR, национальные нормативы каждой страны.

▸ Что они сделали

Kaluzny показал, как Claude обрабатывает поток первичных обращений пациентов до того, как обращение попадает к врачу.

Архитектура такая. Пациент пишет в Doctolib через приложение или сайт. Claude читает обращение, выделяет ключевые симптомы и обстоятельства, классифицирует уровень срочности (плановый визит / приоритетная запись / срочно скорая помощь). К врачу обращение поступает уже с подготовленным резюме: что у пациента, что важно учесть, есть ли в анамнезе что-то релевантное (если пациент даёт согласие на использование истории).

Это не диагностика. Это не замена врача. Это снижение времени, которое врач тратит на чтение и сортировку первичной информации.

Регуляторная архитектура: данные пациента физически не покидают периметра Doctolib (это критично для GDPR и национальных медицинских регуляторов). Claude работает в режиме, аналогичном MCP Tunnel (см. Раздел 2), с гарантией не-выноса данных в облако Anthropic.

▸ Цифры

[К заполнению из записи доклада: сокращение времени врача на одно обращение, рост пропускной способности клиники, удовлетворённость пациентов.]

Что повторимо для российского СМБ

Прямая аналогия для российского рынка: любая компания, которая обрабатывает поток клиентских обращений (входящие звонки, заявки с сайта, обращения в чат), может применить тот же шаблон.

Архитектурная схема. Обращение клиента -> Claude читает -> Claude классифицирует по срочности и теме -> Claude выделяет ключевые факты -> Сотруднику обращение приходит с готовым резюме и приоритетом.

Что это даёт. Сотрудник тратит на одно обращение 30-50% меньше времени (не нужно читать всё и сортировать вручную). Срочные кейсы не теряются в общем потоке (Claude выделяет их в приоритет). Накапливается статистика по типам обращений, которая раньше требовала ручной аналитики.

Кейс #4. Delivery Hero, операционные процессы

Спикер: Rodrigue Schäfer, Delivery Hero **Сессия:** «Building AI-native at enterprise scale» **Track:** Claude Platform, Main stage

Контекст компании

Delivery Hero это группа компаний по доставке еды. Включает Foodpanda, Talabat, Pedidos Ya и другие бренды. Публично на 2024-2025: 1.5+ миллиарда заказов в год, 70+ стран, оценка компании 5-7 миллиардов евро.

Операционная сложность: тысячи курьеров, десятки тысяч ресторанов, миллионы заказов в день. Каждая ошибка в маршрутизации, прогнозе спроса, обработке жалобы это конкретные деньги (испорченные продукты, недовольный клиент, потерянная репутация).

Что они сделали

Schäfer показал три применения Claude в операционной работе.

Маршруты курьеров. Claude помогает оптимизировать распределение заказов между курьерами с учётом погодных условий, текущей загрузки, исторических паттернов. Это не «AI решает за диспетчера», это «AI предлагает диспетчеру варианты, диспетчер выбирает».

Прогноз спроса. Claude анализирует данные о заказах, погоде, событиях в городе (концерты, спортивные матчи, праздники) и предсказывает всплески спроса в конкретных районах. Это помогает заранее увеличить число курьеров в нужных зонах.

Обработка нестандартных жалоб. Стандартные жалобы решаются обычными процессами компенсации. Нестандартные (сложные кейсы с потенциалом эскалации) Claude помогает оператору обработать: предлагает варианты ответа, проверяет историю клиента, оценивает риск ухода клиента к конкуренту.

Опять же, ни одно из применений не «заменяет операционистов». Каждое снимает нагрузку с самых трудоёмких классов задач.

▸ Цифры

[К заполнению из записи доклада: процент оптимизированных маршрутов, точность прогноза спроса, снижение времени обработки сложных жалоб.]

▸ Что повторимо для российского СМБ

Прямая аналогия: любая компания с операционной сложностью (логистика, retail с многими точками, услуги с распределёнными исполнителями) может применить тот же шаблон точечно.

Не пытайтесь сразу «оптимизировать всю операционку через AI». Возьмите одну болевую точку, где сотрудник тратит больше всего времени и совершает больше всего ошибок. Встройте туда Claude как помощника (не как замену). Через 1-2 месяца измерьте эффект, переходите к следующей точке.

Кейс #5. Legora, legal-агенты, унаследовавшие архитектуру code-агентов

Спикер: Jakob Emmerling, Legora **Сессия:** «What legal agents inherit from coding agents: Lessons from Legora» **Track:** Claude Platform, Main stage

▸ Контекст компании

Legora это платформа AI-агентов для юридических фирм. Базируется в Швеции, обслуживает международные юр-фирмы и in-house legal-команды в крупных компаниях. Публично известна тем, что среди клиентов есть несколько Magic Circle юридических фирм (Лондон) и Vault 100 фирм (США).

▸ Главный технический тезис

Архитектура AI-агента для написания кода (как в Claude Code, GitHub Copilot, Cursor) построена на трёх принципах.

Принцип 1: модель видит весь контекст проекта. Файлы, история коммитов, тесты, документация. Не отрывочно, а как единое целое.

Принцип 2: модель имеет инструменты для действий. Запустить тест, открыть файл, изменить строку, проверить compile, сделать commit. Каждый инструмент это конкретный API внутри окружения.

Принцип 3: каждое действие отслеживается. Можно откатиться, можно показать что именно сделал агент, можно подписать изменения.

Emmerling и команда Legora перенесли эти три принципа на работу юриста.

Контекст проекта = все документы клиента. Контракты, переписки, экспертизы, предыдущие судебные решения, базы законодательства.

Инструменты = специализированные интеграции. Поиск по судебной практике. Проверка контрагента в реестрах. Генерация шаблона по типу сделки. Проверка изменений в законодательстве с момента последней работы.

Отслеживание = весь след действий агента можно показать клиенту. Что именно агент сделал. На каких источниках основал вывод. С какой уверенностью.

▸ Почему юристы первыми получили взрыв engagement в Cowork

На той же неделе, что прошла конференция, Anthropic объявил, что legal стал функцией Claude Cowork номер один по engagement среди всех профессиональных доменов. Это не случайность.

Юридическая работа структурно похожа на работу с кодом. Те же признаки.

Большой массив текстов с строгими внутренними связями (договоры с ссылками друг на друга и на законодательство).

Высокая цена ошибки (как и в коде).

Возможность отката (черновик можно переписать, до подписи документ не действителен).

Чёткая система проверки (закон + практика, как в коде тесты + компилятор).

Переноса архитектуры с code-агентов на legal-агентов хватило для того, чтобы получить рабочее решение быстро. В отличие от, например, маркетинга, где архитектура изначально другая (главная ценность это креативность и резонанс с аудиторией, не дисциплина текстуальных связей).

▸ Цифры

[К заполнению из записи доклада: метрики adoption Legora в юридических фирмах, влияние на скорость работы юристов, retention клиентов.]

▸ Что повторимо для российского юриста

Если в вашем юридическом бизнесе есть повторяющиеся задачи (а они есть в каждом), готовая архитектура для их автоматизации уже существует. Не нужно изобретать.

Базовые сценарии, которые работают прямо сейчас на Claude.

Подготовка проекта договора по образцу из вашей базы. Claude читает образец, изменяет под параметры новой сделки, выделяет пункты, требующие индивидуального решения.

Анализ контракта от контрагента. Claude сверяет с вашими шаблонами, помечает изменения, предлагает редактуру. Финальную правку всегда юрист.

Поиск похожих случаев в вашей базе. По 500 предыдущим делам Claude находит 3-5 наиболее близких, подсвечивает релевантные пункты.

Сборка экспертизы для клиента. Claude собирает базовый текст, юрист дописывает уникальные секции и подписывает.

Это не «AI заменит юриста». Это «AI закроет повторяющуюся часть работы юриста, чтобы юрист занимался уникальным».

Бонусный кейс: Spotify devex

Спикер: Niklas Gustavsson, Spotify **Сессия:** «Coding is no longer the constraint: Scaling devex to teams and agents at Spotify» **Track:** Research, Main stage

Spotify не делится конкретными цифрами по количеству разработчиков и эффективности (это коммерческая тайна), но Gustavsson чётко обозначил структурный сдвиг.

В 2024 году ограничивающим ресурсом инжиниринг-команды было «сколько кода вы можете написать». В 2026 году ограничивающим ресурсом стало «сколько идей вы можете протестировать». Claude сократил время от идеи до prototype в работающем продукте в разы. И теперь главный bottleneck не разработка, а целеполагание и приоритизация.

Это контр-нарратив к мейнстриму «AI заменяет программистов». Реальный сдвиг иной: AI меняет роль разработчика. Из «человека, который пишет код» в «человека, который ставит правильные задачи и проверяет результат». Это другой тип работы, с другими навыками, и не каждый разработчик успешно через этот сдвиг проходит.

Для российского СМБ: если в вашей команде есть разработчики, обсудите с ними этот сдвиг. Те, кто примет новую роль, ускорят компанию в разы. Те, кто будет держаться за «пишу код руками», станут узким местом.

Бонусный кейс: Lovable vibecoding в проде

Спикер: Fabian Hedin, Lovable **Сессия:** «How Lovable vibecodes production software at scale» **Track:** Claude Platform, Main stage

Lovable это no-code платформа для создания приложений в режиме «опиши что нужно, получи рабочий продукт». В сообществе разработчиков подход называется «vibecoding»: вы не пишете код, вы описываете желаемое поведение, AI собирает код.

Для большинства СМБ-задач этот режим избыточен (вам не нужно строить приложение, вам нужно использовать существующие). Но как пример того, куда движется AI-разработка, кейс полезен.

Hedin показал, что vibecoding работает в проде ровно тогда, когда у вас есть три вещи: чёткое описание целевого пользовательского сценария, набор готовых интеграционных блоков (auth, payments, storage), и механизм отката изменений. Без этих трёх vibecoding скатывается в хаос недописанного и ломающегося кода.

Структурный вывод: vibecoding это не «AI пишет за меня код», это «AI собирает архитектуру из проверенных блоков по моему описанию задачи». Это разные вещи. Первое нестабильно. Второе работает.

Для российского СМБ применение Lovable или аналогов имеет смысл, когда у вас есть простая задача из стандартного класса (внутренний инструмент, прототип, MVP) и нет ресурсов на классическую разработку. В сложных production-сценариях vibecoding пока не готов.

Сводный паттерн пяти кейсов

Если посмотреть на пять основных кейсов (без бонусных Spotify и Lovable), все они построены вокруг одного принципа.

AI не заменяет функцию целиком. AI вставляется в конкретное место workflow, где сотрудник тратит максимум времени на повторяемые действия, и где ошибка имеет ограниченную цену.

Это применимо к monday.com (Claude внутри workflow, не chat сбоку). К Doctolib (Claude в первичной обработке обращений, не в диагностике). К Delivery Hero (Claude в трёх точечных операциях, не в «оптимизации операционки»). К Legora (Claude в подготовке документов, финальную правку всегда юрист).

И к гипергросту 1→80 это применимо тоже. Claude не заменяет разработчиков, он снимает рутину онбординга и code review, освобождая senior-инженеров для архитектурных решений.

Для российского СМБ это и есть рабочая формула. Не «давайте сделаем AI-стратегию». А «давайте найдём 3-5 мест, где можно вставить Claude точечно, и проверим что работает».

05

ОТ 10 МИНУТ ДО АРХИТЕКТУРНОГО РЕШЕНИЯ

Восемь практик на эту неделю

НА ЭТУ НЕДЕЛЮ

Восемь практик.
От 10 минут
до архитектурного решения.

10 МИНУТ

01
Routines

Запланированные задачи Claude.
Pro 5/день, Max 15/день.

Утренний отчёт → Telegram

30 МИНУТ · ПРЕМИУМ

02
MCP Tunnel

Подключение к CRM/ERP без выноса данных за периметр.
Решает 152-ФЗ, банковскую и врачебную тайну.

Архитектура одобрена ИБ. Возражение «не отдадим данные» закрыто.

1 ВЕЧЕР · ФУНДАМЕНТ

03
Память между сессиями

CLAUDE.md в корне + папка memory/.
people · projects · context · procedures.

Claude запоминает ваши решения за полгода. Не chat-окно.
Соответствует паттерну 5 (stateful agent) из Раздела 3.

2 ЧАСА

04
Proactive workflow

Агент сам инициирует

1 ЧАС

05
Промт = спецификация

Контекст + примеры + цель

1 ЧАС · АНАЛИТИКА

06
Выбор модели

Бенчмарк 10-20 кейсов
на Sonnet · Opus · Haiku.

Sonnet = чаще правильный.
Opus = критичные кейсы.
Haiku = большой объём.

1 ДЕНЬ · СИСТЕМА

07
От prompt к production

Четыре стадии: prompt → scale → hardening → monitoring.
Три метрики: % успеха, стоимость, время.

Метод Дэна Кэри. Code w/ Claude London 19.05.

СТРАТЕГИЧЕСКИЙ ШАГ

08
Калибровка мышления

Класс А: вы решаете, Claude разворачивает варианты.
Класс Б: Claude синтезирует, вы проверяете на здравый смысл.
Класс В: Claude исполняет автономно.

ТЕМП ОСВОЕНИЯ

Одна практика в неделю.

За два месяца у вас собирается полная система.

Уровень, доступный примерно 5% русскоязычной аудитории.

ВИЗУАЛЬНАЯ ШКАЛА СЛОЖНОСТИ

Этот раздел построен как набор готовых к применению практик. Каждая практика разбита по одной схеме: что это, под какую задачу подходит, как настроить, что проверить через неделю. Восемь практик расположены в порядке нарастания сложности: первая работает у любого Pro-пользователя за 10 минут, последняя требует осознанной архитектурной работы.

Применить все восемь сразу не нужно. Возьмите одну-две, которые ближе к вашей текущей боли, доведите их до результата, через неделю возвращайтесь за следующими.

Практика #1. Routines за 10 минут

Что это. Запланированные задачи в Claude Code, которые работают по расписанию без вашего участия. Появились 19 мая 2026 года. Лимиты: 5 routines в день на Pro, 15 на Max, 25 на Team и Enterprise.

Под какую задачу. Любое регулярное действие, которое вы или сотрудник повторяете чаще двух раз в неделю.

Типовые примеры. Каждое утро в 7:00 сборка упоминаний бренда за сутки и отправка в почту. Каждый понедельник в 9:00 сверка остатков в 1С с продажами на маркетплейсе. Каждый вечер в 18:00 короткое summary дня из писем ключевых клиентов. Каждый четверг проверка обновлений в законодательстве по вашей отрасли.

Как настроить. Откройте Claude Code на web, идите в раздел Routines, нажмите Create. Опишите задачу на русском или английском: что делать, какие источники, куда отправить результат, в какое время. Сохраните. Дальше работает само.

Что проверить через неделю. Получили ли вы результат каждый из заявленных дней. Если хотя бы один пропущен, идите в логи routine, смотрите причину. Обычно это лимит запросов или сбой источника данных.

Совет. Не пытайтесь сразу автоматизировать сложный процесс с тремя ветвлениями. Начните с простого: одна задача, один источник, один получатель. Когда заработает стабильно, наращивайте сложность.

Источник: сессия Ralph Ramos «What’s new in Claude Code» (Code w/ Claude London 19.05).

Практика #2. MCP Tunnel за 30 минут

Что это. Зашифрованное исходящее соединение между Claude и вашей внутренней системой (CRM, ERP, база). Внутри вашего периметра ставится маленький шлюз, он сам инициирует соединение к Anthropic. Никаких открытых портов на firewall. Запущен 19 мая 2026 в режиме research preview.

Под какую задачу. Любая интеграция Claude с системами, данные которых не должны покидать периметр компании. 152-ФЗ, банковская тайна, врачебная тайна, коммерческая тайна.

Как настроить. Идите на console.anthropic.com, в раздел Managed Agents, создайте нового агента. Выберите опцию MCP Tunnel. Получите установочный пакет (Docker-образ или Linux-бинарь). Передайте админу. Админ ставит шлюз внутри сети, где есть доступ к нужной системе. Пропишите в шлюзе локальные адреса систем (вида crm.local или 10.0.1.5). Запустите шлюз. На стороне Anthropic появится отметка «канал готов». В настройках агента укажите, что он работает через этот туннель.

По времени. 30-60 минут, если у вас есть админ. Без админа: запросите у внутреннего IT.

Что проверить через неделю. Что соединение стабильное (не разваливается при сетевых перебоях). Что аудит-логи шлюза собираются и хранятся. Что ИБ-офицер видел архитектурное описание и не имеет возражений.

Caveat. Research preview, не GA. Для пилотов, R&D, внутренних сценариев работает. Для production-критичной нагрузки с финальным переходом дождитесь GA-релиза (ожидается осенью 2026).

Источник: Anthropic Help Center + сессии «How to get to production faster with Claude Managed Agents» (Michael Cohen, Harrison Stall), «Getting more out of the Claude Platform» (Punit Shah).

Практика #3. Картирование внутренней памяти

Что это. Создание системы памяти, которая живёт между сессиями работы с Claude. Соответствует паттерну 5 из Раздела 3 (stateful agent).

Под какую задачу. Любая работа, где вы хотите, чтобы Claude помнил ваши решения, ваш стиль, ваши предыдущие проекты, а не начинал каждую сессию с нуля.

Как настроить. Создайте в корне рабочей папки файл `CLAUDE.md`. Запишите туда: - Кто вы и чем занимаетесь (1-2 предложения). - Ваш регистр работы (формальный, разговорный, технический). - Ваши регулярные процессы (что повторяется еженедельно, ежемесячно). - Ваши принципиальные «нет» (что вы НЕ делаете в работе). - Ваши предпочтения по формату ответа (длина, структура).

Создайте папку `memory/` с подпапками: - `people/`. Профили ключевых людей в ваших проектах. - `projects/`. Детали активных проектов. - `context/`. Кристаллизованные принципы и инсайты (растут со временем). - `procedures/`. Повторяемые процедуры.

Каждый раз, когда говорите Claude «запомни X», он дописывает это в нужный файл.

Что проверить через неделю. Что Claude в новой сессии помнит вас без переобъяснений. Что он сам обращается к вашим записям при принятии решений. Что папка `memory/` растёт органично, без необходимости вашего ручного редактирования.

Совет. Не пытайтесь сразу описать всё. Начните с `CLAUDE.md` на одну страницу и одной подпапки в `memory/`. Доращивайте по мере необходимости.

Источник: сессия Ravi Trivedi «Memory and dreaming for self-learning agents» + годовой опыт автора с этой архитектурой.

Практика #4. Proactive workflow

Что это. Архитектура, в которой агент сам инициирует действие на основе наблюдаемых сигналов, не ждёт команды.

Под какую задачу. Мониторинг состояний, где вам важно среагировать на изменение быстро, но вы не можете постоянно проверять вручную.

Типовые примеры. Падение конверсии в продаже на 20% за день. Появление негативного отзыва на маркетплейсе. Заявка от ключевого клиента вне рабочих часов. Изменение цены поставщика. Обновление в законодательстве по вашей отрасли.

Как настроить. Используя Routines (Практика #1) как основу, опишите задачу не как «делай X в Y:00», а как «следи за условием Z и сделай X если условие сработало».

Пример: «Каждый час проверяй упоминания нашего бренда в новостных источниках. Если найдёшь упоминание с негативной тональностью, пришли мне в Telegram сообщение со ссылкой и краткой оценкой».

Что проверить через неделю. Что агент срабатывает на реальные сигналы (не пропускает важное). Что агент НЕ срабатывает на ложные сигналы (false positives). Если ложных срабатываний больше 20%, уточняйте формулировку условия.

Совет. Начните с проактивного агента на низком риске. Уведомление в Telegram это безопасный выход. Не давайте проактивному агенту прав отправлять что-то от вашего имени без подтверждения.

Источник: воркшоп Maya Nielan «Build a proactive agent workflow with Claude Code».

Практика #5. Промтинг как спецификация

Что это. Промт это не команда, а спецификация. Если вы пишете «напиши пост», вы получаете средний пост. Если вы пишете спецификацию (контекст + примеры + цель + критерии), вы получаете результат, который попадает в эти критерии.

Под какую задачу. Любая регулярная задача с Claude, где результат имеет повторяющиеся требования к качеству и формату.

Как настроить. Заведите файл `prompts/` или раздел в `CLAUDE.md` с шаблонами промтов. Для каждой регулярной задачи опишите шаблон по четырём блокам.

Блок 1, контекст. Кто целевая аудитория, на каком языке, в каком регистре, с каким бэкграундом.

Блок 2, примеры. 3-5 примеров ваших лучших результатов в этой задаче. Не описания «как должно быть», а реальные образцы. Это самая важная часть. Claude по примерам понимает ваш стиль точнее, чем по любым правилам.

Блок 3, цель. Что должно произойти в голове или поведении читателя/пользователя после взаимодействия с результатом.

Блок 4, критерии. По чему вы будете оценивать что задача выполнена хорошо. Конкретно. «Убедительно» не критерий. «Дочитают до конца 60% открывших» это критерий.

Что проверить через неделю. Что вы тратите на каждую задачу меньше времени, чем неделю назад. Что результаты с первого раза попадают в нужный регистр.

Источник: сессия Margot van Laar «The prompting playbook».

Практика #6. Выбор модели под задачу

Что это. Тестирование нескольких моделей на вашей конкретной задаче с целью выбрать ту, что даёт лучший результат при разумной цене.

Под какую задачу. Любая регулярная задача с Claude, которую вы планируете масштабировать. Пока задача редкая и разовая, выбор модели не критичен. Когда задача станет ежедневной, разница между правильной и неправильной моделью это сотни тысяч рублей в год.

Как настроить. Подготовьте бенчмарк-набор из 10-20 типичных входов вашей задачи (с уже известными правильными результатами).

Запустите задачу на трёх моделях. Базовая: Claude Sonnet (или эквивалент). Премиум: Claude Opus. Экспресс: Claude Haiku.

Оцените три параметра.

Качество результата. По вашим критериям (см. Практику #5). Каждый из 10-20 кейсов оценивается 0/1 (попал/не попал) или по 5-балльной шкале.

Скорость. Среднее время на один кейс.

Стоимость. Реальная цена API-запроса умножить на ожидаемое количество кейсов в месяц.

Выберите модель не по «которая умнее», а по «которая даёт нужное качество при приемлемой цене и скорости».

Что проверить через неделю. Что вы знаете точную годовую стоимость задачи на выбранной модели. Что вы знаете, какой кейс модель валит чаще всего, и есть ли смысл его обрабатывать отдельно (например, более дорогой моделью или человеком).

Совет. Для большинства задач Sonnet это правильный выбор. Opus оправдан только там, где качество критично и кейсов немного. Haiku оправдан только там, где кейсов много и качество терпимое.

Источник: воркшоп Lucas Smedley «Picking the right model».

Практика #7. Designing с Claude: от prompt к production

Что это. Дисциплина перевода промта в production-готовый процесс. Не «получилось разок» и не «выложили скрипт», а «работает стабильно, метрики собираются, ошибки ловятся».

Под какую задачу. Любая задача, которая прошла стадию proof-of-concept и которую вы хотите запустить в регулярную работу.

Как настроить. Пройдите четыре стадии.

Стадия 1, prompt. Получите рабочий результат на одном кейсе. Это базовая стадия, обычно занимает 1-3 часа.

Стадия 2, scale. Прогоните этот промт через 10-20 типичных кейсов. Соберите статистику: сколько прошло, сколько провалилось, на каких именно входах. Это даёт первое представление о реальном качестве.

Стадия 3, hardening. Для каждого типа провала из стадии 2 решите: фикс в промте, обработка исключения в коде, или передача человеку. Не пытайтесь починить всё в промте, это путь к 50-страничным промтам, которые сами по себе становятся источником ошибок.

Стадия 4, monitoring. Запустите задачу в проде с автоматическим сбором метрик. Минимальный набор: процент успешных выполнений, средняя стоимость одного выполнения, среднее время. Эти три метрики смотрите еженедельно.

Что проверить через неделю. Что метрики собираются автоматически. Что вы знаете порог, ниже которого нужно вмешиваться (например, если процент успешных упал ниже 90%, останавливаем и разбираемся).

Источник: сессия Dan Cary «Designing with Claude: From prompt to production».

Практика #8. Калибровка распределения мышления

Что это. Осознанное решение, какую часть мыслительной работы вы оставляете себе, а какую отдаёте Claude. И где между ними находится точка опоры.

Под какую задачу. Стратегический шаг для тех, кто работает с Claude больше месяца и понимает, что простое «давай Claude всё делает» приводит либо к потере собственного мышления, либо к плохому качеству решений.

Как настроить. Разделите свои регулярные задачи на три класса.

Класс А, мышление. Задачи, в которых главная ценность это качество вашего мышления: стратегия, отношения, ключевые решения, творческие выборы. Эти задачи Claude помогает развернуть (генерация альтернатив, проверка логики), но решает их человек.

Класс Б, синтез. Задачи, в которых главная ценность это переработка большого объёма информации в компактный вывод: research, summary, аналитика. Эти задачи Claude делает сам, человек проверяет результат на здравый смысл и принимает на основе вывода.

Класс B, исполнение. Задачи, в которых главная ценность это аккуратное выполнение по чёткому процессу: оформление документов, отправка отчётов, рутинная коммуникация. Эти задачи Claude делает полностью автономно (после первой настройки).

Заведите эту классификацию в `CLAUDE.md`. Для каждой регулярной задачи помечайте класс.

Что проверить через неделю. Что вы не путаете классы. Что для класса A вы не позволяете Claude принимать решение за вас. Что для класса B вы не теряете время на проверку каждого исполнения.

Источник: доклад Alexander Bricken «The thinking lever».

Сводная таблица практик

Восемь практик распределены по уровням сложности и срокам внедрения.

Уровень 1, за 10-30 минут на практику. #1 Routines, #6 Выбор модели.

Уровень 2, за 1-2 часа на практику. #2 MCP Tunnel, #4 Proactive workflow, #5 Промтинг как спецификация.

Уровень 3, за день или больше. #3 Картирование памяти, #7 От prompt к production, #8 Калибровка распределения мышления.

Целевой темп освоения: одна практика в неделю. За два месяца у вас собирается полная система. Раньше освоить все восемь технически возможно, но без оседания привычки практика остаётся декларативной, не операционной.

После всех восьми практик вы будете работать с AI на уровне, который сегодня доступен примерно 5% русскоязычной аудитории. И при этом каждая отдельная практика занимает реалистичные часы, не месяцы.

06

КОНТР-НАРРАТИВ АВТОНОМИИ

Stop babysitting your agents

Шкала уровней автономии AI-агента

Три уровня риска. Три режима автономии. Прямая связка с runtime patterns.

Адаптация воркшопа Sid Bidasaria «Stop babysitting your agents», Code w/ Claude London 19.05.2026

НИЗКИЙ РИСК

До 10 000 руб. ущерба за операцию

РЕЖИМ АВТОНОМИИ

Автономия без надзора

Агент работает.
Человек делает выборочный аудит
раз в неделю или реже.

ПРИМЕРЫ ЗАДАЧ

- Черновик email
- Заметки в CRM
- Summary встречи
- Первичная категоризация
- Помощник в копирайтинге

ПОДХОДЯЩИЕ ПАТТЕРНЫ

1. Чистый chat
2. Tool-augmented chat
5. Stateful agent

КАК ПРОВЕРИТЬ ЧЕРЕЗ НЕДЕЛЮ

- ✓ Сколько случаев агент обработал
- ✓ Сколько часов человека высвободил
- ✓ Выборка из 10 случаев — качество
- ✓ Если > 90% OK — продолжайте

СРЕДНИЙ РИСК

10 тыс. — 1 млн руб. ущерба

РЕЖИМ АВТОНОМИИ

Автономия с аудитом

Агент исполняет автоматически.
Человек смотрит выборку
из 10-20 случаев в неделю.

ПРИМЕРЫ ЗАДАЧ

- Обработка заказов с маркетплейса
- Ответы клиентам по обращениям
- Базовая аналитика для решений
- Подготовка отчётов
- Маршрутизация задач между людьми

ПОДХОДЯЩИЕ ПАТТЕРНЫ

3. ReAct-агент
4. Plan-and-execute
5. Stateful с аудитом памяти

КАК ПРОВЕРИТЬ ЧЕРЕЗ НЕДЕЛЮ

- ✓ Процент успешных выполнений
- ✓ Стоимость одного выполнения
- ✓ Среднее время на кейс
- ✓ Дрейф качества по статистике

ВЫСОКИЙ РИСК

Более 1 млн руб. или необратимо

РЕЖИМ АВТОНОМИИ

Human-in-the-loop

Агент готовит.
Человек подписывает
каждое отдельное действие.

ПРИМЕРЫ ЗАДАЧ

- Перевод денег
- Документ внешнему контрагенту
- Медицинские решения
- Финансовая отчётность регулятору
- Юридические позиции в споре

ПОДХОДЯЩИЕ ПАТТЕРНЫ

4. Plan-and-execute + confirm
6. Multi-agent с эскалацией

Точка подтверждения перед
каждым необратимым шагом обязательна

КАК ПРОВЕРИТЬ ЧЕРЕЗ НЕДЕЛЮ

- ✓ Все точки подтверждения работают
- ✓ Время человека на подпись
- ✓ Качество подготовки агентом
- ✓ Ноль необратимых ошибок

Без матрицы выбор «нянчить — не нянчить» это монетка. С матрицей это инженерное решение.

Under the Hood — адаптация для русскоязычного СМБ

Два мейнстрима о надзоре за AI

В 2025-2026 годах сложились две противоборствующие позиции о том, как использовать AI-агентов в работе.

Первая позиция, доминирующая в корпоративной среде и в большинстве публикаций для регулируемых отраслей: AI требует постоянного человеческого надзора. Каждое действие агента должно быть проверено человеком, каждый вывод подтверждён, каждое решение визировано. Это позиция аудита, ИБ, юристов. Она опирается на риск-фрейм: AI может ошибаться, поэтому нужен страховочный пояс из людей.

Вторая позиция, доминирующая среди AI-энтузиастов и в стартапах: AI должен работать автономно, иначе теряется весь смысл автоматизации. Если вы стоите за плечом каждого исполнения, вы платите за модель, но получаете эффект ассистента уровня стажёра. Зачем это? Это позиция эффективности. Она опирается на ROI-фрейм: автоматизация не работает там, где человек дублирует машину.

Между этими двумя позициями есть третий путь, и он гораздо точнее обоих. Этот путь обозначил Sid Bidasaria в своём воркшопе Stop Babysitting Your Agents на Code w/ Claude London 19 мая 2026 года.

Что говорит Sid

Тезис воркшопа звучит провокационно: если вы постоянно надзираете за агентом, вы используете его неправильно. Это не отказ от безопасности и не призыв запустить агента и забыть про него. Это требование осознанного выбора уровня автономии под уровень риска задачи.

Нянчить агента это плохая методология. Время человека, потраченное на проверку каждого вывода, обнуляет выигрыш от автоматизации. Если вы автоматизировали обработку 1000 заказов в день и проверяете каждый, вы по-прежнему обрабатываете 1000 заказов в день вручную, только медленнее, чем без агента.

Доверять без калибровки тоже плохая методология. Если вы запустили агента в продакшен и не возвращаетесь посмотреть, что он делает, это вопрос времени, когда он сожжёт что-то ценное. Не «может сожжёт», а сожжёт. AI-агенты в 2026 году не дают гарантии корректности на длинных временных горизонтах. Они дают статистическую корректность, и статистика рано или поздно реализуется в виде ошибки.

Правильная методология звучит так: выбрать паттерн runtime-архитектуры под уровень риска. Это конкретное инженерное решение, не философский выбор между «контроль» и «свобода».

Три уровня риска и три режима автономии

Sid в воркшопе предложил трёхуровневую градацию.

Низкий риск: автономия без надзора

Задачи, где цена ошибки ограничена и обратима. Черновик email, заметки в CRM, summary встречи, первичная категоризация входящих запросов. Если агент ошибся, последствия исправляются за минуты, без потери денег и репутации.

Для таких задач правильный режим: автономия без человеческого надзора. Агент работает, периодически вы делаете выборочный аудит результатов, реальный контроль происходит на уровне статистики, не на уровне отдельных решений.

Соответствующие паттерны: 1 (чистый chat) или 2 (tool-augmented chat) для задач с одношаговым выводом. Паттерн 5 (stateful agent) для задач с накоплением знаний без критичной цены ошибки.

Средний риск: автономия с регулярным аудитом

Задачи, где цена ошибки заметна, но не критична. Обработка заказов из маркетплейса с реальными деньгами в небольших суммах. Ответы клиентам по обращениям без сложных юридических нюансов. Базовый анализ данных для внутренних решений.

Для таких задач правильный режим: автономное исполнение, но с регулярным человеческим аудитом результатов. Не каждое действие, но раз в неделю или раз в месяц человек смотрит на статистику и на случайную выборку из 10-20 случаев. Цель аудита: ловить дрейф качества, обнаруживать новые edge cases, накапливать обратную связь для уточнения промпта или дообучения.

Соответствующие паттерны: 3 (ReAct) или 4 (plan-and-execute) для автономного исполнения. Паттерн 5 (stateful) для задач, где аудит ведётся по накопленному опыту.

Высокий риск: human-in-the-loop

Задачи, где цена ошибки высокая и трудно или невозможно обратимая. Перевод денег. Отправка документа внешнему контрагенту, у которого этот документ становится формальным обязательством. Принятие решения по медицинскому случаю. Финансовая отчётность для регулятора. Юридическая позиция в споре.

Для таких задач правильный режим: human-in-the-loop. Агент готовит, человек подписывает. Это не «человек проверяет каждое слово», а «человек проверяет ключевые точки решения и подписывает результат». Агент остаётся полезен, потому что снимает с человека рутинную часть подготовки. Человек остаётся в контуре, потому что только он может нести ответственность за необратимое действие.

Соответствующие паттерны: 4 (plan-and-execute) с обязательной точкой подтверждения перед исполнением. Паттерн 6 (multi-agent с координатором), где координатор поднимает на человека вопросы за пределами своей зоны компетенции.

Связка с шестью паттернами

Главная мысль Sid и причина, по которой эта сессия в одном ряду по важности с самим академическим документом arXiv 2605.20173:

Выбор режима автономии это не отдельная политическая или операционная проблема. Это прямое следствие выбора runtime-паттерна. Каждый из шести паттернов из Раздела 3 имплицитно задаёт уровень автономии, который для него работает.

Чистый chat (паттерн 1) живёт в режиме автономии без надзора по своей природе: он не делает действий, он только говорит.

Tool-augmented chat (паттерн 2) живёт в гибридном режиме: для информационных запросов автономия без надзора, для вызова инструментов с побочными эффектами желателен предварительный confirm.

ReAct (паттерн 3) и plan-and-execute (паттерн 4) предполагают регулярный аудит для среднего риска и обязательный human-in-the-loop для высокого. Без явной точки подтверждения они опасны.

Stateful agent (паттерн 5) требует периодического аудита накопленного состояния: данные могут дрейфовать, паттерны устаревать, и без регулярной ревизии агент может тихо начать работать неправильно.

Multi-agent с координатором (паттерн 6) требует chỉ четкой эскалации: координатор должен знать, когда поднимать вопросы на человека, и эта точка эскалации должна быть формализована в архитектуре.

Практический чеклист

Перед запуском любого AI-агента в проде ответьте на три вопроса.

Вопрос 1. Какой максимальный денежный или репутационный ущерб от одной неправильной автономной операции? Если меньше 10 000 рублей или эквивалента: низкий риск. От 10 000 до миллиона: средний. Выше миллиона или невосполнимая репутационная потеря: высокий.

Вопрос 2. Обратимо ли неправильное действие? Если да, в течение какого времени и за чьи деньги. Полностью обратимо за минуты: можно автономно. Обратимо с потерями: с аудитом. Необратимо: только human-in-the-loop.

Вопрос 3. Какая частота операций? Если 1-10 в день: имеет смысл проверять каждое. Если 100-1000: имеет смысл автономия с аудитом. Если 10 000+: автономия без выборочного надзора, потому что физически нельзя проверить вручную.

Сложите ответы в матрицу. Каждая клетка матрицы это конкретный runtime-паттерн из Раздела 3 и конкретный режим автономии из этого раздела. Без матрицы выбор между «нянчить» и «не нянчить» это монетка. С матрицей это инженерное решение.

Что НЕ говорит Sid

В воркшопе не говорилось, что любой агент должен быть автономным. Не говорилось, что надзор это всегда плохо. Не говорилось, что AI всегда лучше человека. Это важно подчеркнуть, потому что в Twitter-обзорах сессии тезис огрубили до «не контролируйте AI». Этот огрублённый тезис ложен.

Реальный тезис: контролируйте AI ровно настолько, насколько требует риск задачи. Не больше. Не меньше. Чтобы определить ровно сколько, у вас должна быть формальная градация рисков и формальный выбор runtime-паттерна.

Это инженерная работа. Её можно сделать за день для всего портфеля AI-задач компании. После этого ваши сотрудники перестанут нянчить агентов, и одновременно перестанут отпускать их в свободное плавание там, где этого делать нельзя.

Дальнейшее чтение

- Запись воркшопа Sid Bidasaria «Stop babysitting your agents», Code w/ Claude London 19.05.2026
- Раздел 3 этого PDF «Карта шести runtime patterns»
- arXiv 2605.20173 «A Methodology for Selecting and Composing Runtime Architecture Patterns for Production LLM Agents»
- arXiv 2605.10481 «Safe Multi-Agent Behavior Must Be Maintained, Not Merely Asserted: Constraint Drift in LLM-Based Multi-Agent Systems»

07

ТОЛЬКО ДЛЯ UNDER THE HOOD

Что Anthropic строит на 18 месяцев

ШУЛЕРСКАЯ ОПТИКА

Пять уровней игры

Anthropic играет на пятом. Конкуренты застряли на первом-четвёртом.

ГЛАВНОЕ ОТКРЫТИЕ

Anthropic не борется за корону.
Anthropic пишет архитектуру правил.

По этим правилам через два года будет определяться, кто чему король.

Архитектор важнее короля.

КАРТА JOKER ВНЕ КОЛОДЫ

Anthropic сложил Joker (AI) + Vatican-tag (легитимность 2000 лет).
Метакомбинация, которую невозможно повторить.

Ни за деньги. Ни за пять лет работы. Карта одноразовая.

Карта стратегии Anthropic, прочитанная с расстояния

24 сессии в Лондоне это не случайная подборка. Это композиция, в которой Anthropic подсвечивает то, во что вкладывает следующие 18 месяцев. По сессиям можно построить карту направления, как по фрагментам мозаики видно общий рисунок.

В этом разделе мы реконструируем эту карту через четыре линии: коммерческую (как Anthropic зарабатывает), стратегическую (с кем и против кого играет), методологическую (какой язык навязывает индустрии), и культурную (где ищет легитимность).

Это раздел не для широкой аудитории. Здесь авторская интерпретация, не объективный пересказ. Кто-то прочитает её как точную карту, кто-то как одну из возможных. Важно понимать: на горизонте 18 месяцев таких карт может быть несколько, и они все будут работать как линзы для принятия собственных решений.

Линия 1. Vertical-as-stack против universal model

Главный коммерческий ход Anthropic 2026 года это переход от «универсальной модели для всех» к «вертикальному стеку для регулируемых отраслей». Сессии в Лондоне это документируют явно.

Vertical-as-stack означает следующее. Anthropic не пытается быть лучшим chatGPT-конкурентом. Anthropic пытается быть единственным выбором для CIO банка, страховой, медицинской платформы, юридической фирмы. Это покупатель с большим бюджетом, длинным циклом продажи, и высоким требованием к комплаенсу.

В Лондоне это видно так. Self-Hosted Sandboxes (см. Раздел 2) это инструмент для тех, кому не пройти ИБ с обычным облаком. MCP Tunnels это инструмент для тех, чьи данные нельзя выпускать наружу. 20+ legal MCPs это полная экосистема под одну вертикаль (юристы). Кейс Doctolib и Legora это два конкретных примера, где архитектура работает в регулируемых отраслях.

OpenAI идёт другой дорогой. Universal model: GPT-5.5 для всех, ChatGPT для всех, Codex для всех разработчиков. Это путь массового рынка. У OpenAI 800+ миллионов еженедельных активных пользователей по публичным данным на конец 2025 года. У

Anthropic порядок меньше. Но у Anthropic выручка приходит из крупного enterprise, где средний контракт измеряется сотнями тысяч и миллионами долларов в год, не подпиской за 20.

К концу 2027 года это разделение станет окончательным. OpenAI = массовый рынок. Anthropic = регулируемый enterprise. Google = enterprise на собственной cloud-инфраструктуре. Meta = open-source и собственные продукты. xAI = специальные применения (правительство, оборона).

Для российского предпринимателя структурный вывод. Если ваш бизнес работает в регулируемой отрасли (банки, страховые, медицина, юристы, госуслуги, оборона), Anthropic становится естественным выбором инфраструктуры. Если ваш бизнес работает в массовом продуктовом сегменте (B2C приложения, контент, развлечения), OpenAI остаётся первой опцией. Если ваш бизнес работает в технологическом продакшене (devtools, инфраструктура), Google и Anthropic конкурируют. Если вы сильно зависите от российской регуляторики (госконтракты, ВПК, чувствительные данные), то ни один из этих, и вам в локальные модели (Yandex, GigaChat).

Линия 2. Игра пяти уровней

Чтобы понять, что делает Anthropic, недостаточно смотреть на коммерческую линию. Здесь работает логика пяти игровых уровней, которую полезно проговорить отдельно.

Любая большая корпоративная игра разворачивается на пяти уровнях одновременно.

Уровень 1, прямая выгода. Лучше продукт, больше клиентов, выше выручка. На этом уровне играют все.

Уровень 2, ролевая фасада. PR, образ компании в медиа, репутация. На этом уровне играют через twitter-аккаунты CEO, пресс-релизы, конференции.

Уровень 3, структурная ловушка. Стандарты, протоколы, экосистемы, в которые конкурент должен зайти, чтобы не отстать (и тем самым работать на тебя). На этом уровне играют через open-source, через стандарты IETF/W3C, через приобретения ключевых инфраструктурных компаний.

Уровень 4, нарративная реальность. Какой язык индустрия использует, чтобы говорить о продукте. Кто задаёт термины, тот задаёт направление развития. На этом уровне играют через peer-review, через академические публикации, через выступления на ключевых конференциях.

Уровень 5, метасистема. Кто определяет правила, по которым формулируются сами эти уровни. На этом уровне играют через институциональные альянсы, через регуляторные процессы, через культурную легитимизацию.

OpenAI большую часть 2024-2025 играл на Уровне 1 (лучший продукт) и Уровне 2 (PR через Альтмана). Это эффективная стратегия для массового рынка.

Anthropic в 2026 году играет на всех пяти одновременно, но центр тяжести сместился вверх.

Уровень 1: Claude Opus 4.7 как лучшая модель для задач высокой сложности. Это базовый уровень, где Anthropic держит конкурентность.

Уровень 2: тихое PR-присутствие. Анропик публичен меньше OpenAI, не строит CEO-культ. Это сознательный выбор: меньше шума, меньше уязвимости.

Уровень 3: MCP как стандарт. Anthropic запустил Model Context Protocol в 2024 году как открытый стандарт. К 2026 году MCP стало де-факто протоколом, через который AI-агенты подключаются к внешним системам. Microsoft, Google, OpenAI вынуждены поддерживать MCP. Anthropic держит роль steward стандарта.

Уровень 4: методологический язык. arXiv 2605.10787 (Clean-Slate Bias) и arXiv 2605.20173 (6 runtime patterns) выходят с активным участием авторов из Anthropic-со-общества. Сессии The Thinking Lever, Stop Babysitting, The Capability Curve формируют язык, на котором индустрия будет говорить о методологии. Это не маркетинг, это формирование общего нарратива.

Уровень 5: культурная легитимизация. 25 мая 2026 года папа Лев XIV опубликовал первую папскую энциклику об AI «Magnifica Humanitas». На сцене запуска в Vatican Synod Hall единственный человек из AI-индустрии: Кристофер Олах, со-основатель Anthropic, глава исследований по интерпретируемости. Это уровень, на котором конкуренты вообще не играли.

Сумма всех пяти уровней даёт картину, в которой Anthropic не борется за корону «лучшей модели». Anthropic пишет архитектуру правил, по которым через два года будет определяться, кто чему король.

Линия 3. Методологический слой как продукт

Если посмотреть на 24 сессии в Лондоне через призму «что Anthropic хочет, чтобы индустрия выучила», явно проступает методологический слой как сознательно конструируемый продукт.

Это не «вот наша модель, пользуйтесь». Это «вот наш способ думать о AI, и кто думает в наших терминах, кому проще покупать наши инструменты».

Конкретные элементы методологического слоя.

Шесть runtime patterns. Это язык, на котором можно описать любой AI-агент. Anthropic не изобрёл эти паттерны, но кодифицировал их в академическом документе с участием своих исследователей.

Stop babysitting и три уровня риска. Это язык, на котором можно говорить об автономии AI без скатывания в крайности «нянчить» или «отпустить».

The Thinking Lever. Это язык, на котором можно говорить о распределении мышления между человеком и AI.

The Capability Curve. Это язык, на котором можно калибровать ожидания от модели под текущую фазу её развития.

Memory and Dreaming. Это язык, на котором можно говорить об архитектуре долгой памяти агента.

Prompting Playbook. Это язык, на котором можно говорить о промптинге не как о трюке, а как о спецификации.

Каждый из этих элементов в отдельности звучит как полезная техническая идея. Сумма даёт целый словарь, на котором инженерные команды будут говорить о AI следующие 2-3 года.

И когда команда говорит на этом словаре, естественный выбор инструментов это инструменты Anthropic. Не потому что они объективно лучше (на конкретной задаче иногда хуже), а потому что они построены в той же логике, что и словарь. Это и есть структурная ловушка Уровня 3 в чистом виде.

Линия 4. Культурный hedge через Vatican

25 мая 2026 года, через 6 дней после конференции в Лондоне, папа Лев XIV публикует первую папскую энциклику об AI. На сцене запуска Кристофер Олах, со-основатель Anthropic и глава interpretability research.

Это не отдельное событие. Это финальная скрепа всей стратегии 2026 года.

Энциклика готовится годами. Подпись 15 мая на 135-летие Rerum Novarum (1891, Лев XIII, о труде и капитале) это не совпадение, это композиция. Anthropic 2-3 года работал с Ватиканом, держал людей под темой интерпретируемости с моральным углом, вёл дипломатию.

Что Anthropic получает этим ходом.

В судебных процессах будущего (а они будут, AI-регулирование только разворачивается) у Anthropic появляется аргумент за пределами технического спора. Не «наша модель безопаснее по таким-то метрикам», а «институт, существующий 2000 лет, признал наш подход к безопасности AI как защиту человеческого достоинства».

В корпоративных департаментах комплаенса появляется дополнительный аргумент. CIO банка может сослаться на Vatican-tag в риск-документации. Это не маркетинг, это снижение argumentation cost для риск-менеджмента.

В найме senior researchers появляется моральная опора. Лучшие учёные в интерпретируемости и alignment идут туда, где есть моральная значимость работы. Vatican-tag сделал Anthropic первым выбором.

Почему этот ход не могли сделать конкуренты.

Сэм Альтман и OpenAI: после скандала с увольнением CEO советом директоров в ноябре 2023 года морального капитала, который Ватикан мог бы взять в зачёт, у него нет.

Демис Хассабис и DeepMind: Google слишком агрессивно коммерциализировал AI, чтобы Ватикан выбрал их интерпретируемость как символ.

Янн ЛеКун и Meta: его публичная позиция скептика safety-фрейма теперь стоит против папской критики.

Илон Маск и xAI: политическая токсичность и Grok-инциденты делают невозможной любую серьёзную институциональную связку.

Anthropic оказался единственной кандидатурой не потому что повезло, а потому что 2-3 года работал ровно над тем, чтобы стать единственной кандидатурой.

Что это значит для российского предпринимателя на горизонте 18 месяцев

Сводный прогноз.

К концу 2026 года. Регулируемые отрасли (банки, страховые, медицина, юристы) в США и ЕС начнут массово выбирать Anthropic как первого AI-партнёра. В России процесс пойдёт медленнее из-за геополитической изоляции, но через 6-9 месяцев аналогичный сдвиг произойдёт и в РФ-сегменте.

К середине 2027 года. Методологический язык 6 runtime patterns / Stop Babysitting / The Thinking Lever станет стандартным для тех команд, которые работают с AI серьёзно. Те, кто не выучил этот язык, начнут отставать в найме инженеров и в качестве архитектурных решений.

К концу 2027 года. Появятся первые формальные регуляторные нормы (в ЕС через AI Act дополнения, в США через executive orders), которые будут опираться на терминологию методологического слоя. Это будет первая регуляторная победа Anthropic.

К 2028 году. Игра пяти уровней закончится. Останется 3-4 крупных AI-поставщика, каждый с собственной нишей. Anthropic = регулируемый enterprise + методологический канон. OpenAI = массовый рынок + продуктовая экосистема. Google = облачный enterprise + специализированные исследования. Локальные игроки (Yandex, Sber для РФ, Mistral для ЕС, китайские для Азии) = суверенный рынок.

Для российского предпринимателя точка вхождения на горизонте 18 месяцев такая.

Если вы в регулируемой отрасли: начинайте пилот на Anthropic-стеке с использованием Self-Hosted Sandboxes и MCP Tunnels. Окно 12-18 месяцев, потом конкуренты догонят, и преимущество исчезнет.

Если вы в массовом продукте: оставайтесь на OpenAI или Yandex/Sber, но изучите методологический язык 6 runtime patterns. Это поможет принимать лучшие решения внутри любого стека.

Если вы в обоих сегментах: сделайте multi-vendor стратегию. На AI-агентах с критичными данными Anthropic, на массовых задачах OpenAI или Yandex. Через 12-18 месяцев это станет нормой.

В любом случае: выучите методологический язык. Через год это станет общим словарём индустрии. Кто выучит первым, тот получит преимущество в найме, продажах, и переговорах с регулятором.

Окно 12-18 месяцев. После этого новые правила игры будут уже написаны Anthropic и его союзниками. Останется только следовать им или искать обходные пути в собственных вертикалях.

СТРАТЕГИЧЕСКИЙ РАЗЛОМ 2026

Vertical-as-stack

против

Universal model

ИГРАЕТ НА:

ANTHROPIC

Vertical-as-stack для регулируемых отраслей

ЦА

CIO банка
CFO страховой
COO медицины

КОНТРАКТ

$100K-
$1M+ /год

1. **Self-Hosted Sandboxes (public beta)**
2. **MCP Tunnels (research preview)**
3. **20+ legal MCPs + 12 practice plugins**
4. **Magnifica Humanitas: Vatican-tag**
5. **6 runtime patterns как peer-reviewed канон**

ИГРАЕТ НА:

OPENAI

Universal model для массового рынка

ЦА

Все потребители и developers

СТА

$20
/мес ChatGPT

1. **GPT-5.5 как универсальная модель**
2. **Codex on-prem с Dell (один партнёр)**
3. **800M+ weekly active users**
4. **CEO-публичность через Альтмана**
5. **C2PA + SynthID watermarks**

К КОНЦУ 2027 ГОДА

Разделение станет окончательным.

Каждый игрок в своей нише. Конкуренции по бенчмаркам не будет.

ЧТО ЭТО ЗНАЧИТ ДЛЯ ВАС

Регулируемая отрасль?

152-ФЗ, банковская, врачебная, юр-тайна.
Anthropic становится естественным выбором инфраструктуры.

Массовый B2C продукт без чувствительных данных.
OpenAI остаётся первой опцией. Или локальные Yandex и Sber.

08

ПРИЛОЖЕНИЕ

Карта 24 сессий

24 сессии Code w/ Claude London, 19 мая 2026

Карта по четырём трекам с оценкой релевантности для русскоязычного СМБ

Релевантность для СМБ: Must-watch (13) · Для технарей (9) · Контекст (2)

Research

8 сессий — фундамент и методология

- **04. Picking the right model**
  Lucas Smedley
- **05. Coding is no longer the constraint**
  Niklas Gustavsson, Spotify
- **06. Designing with Claude**
  Dan Cary, prompt to production
- **13. The thinking lever**
  Alexander Bricken
- **17. The capability curve**
  Jeremy Hadfield
- **24. The prompting playbook**
  Margot van Laar

Claude Platform

9 сессий — функции и интеграции

- **03. Memory and dreaming**
  Ravi Trivedi, self-learning agents
- **11. Claude on AWS Bedrock**
  Antonio Rodriguez
- **12. How Lovable vibecodes**
  Fabian Hedin, production at scale
- **15. AI-native at enterprise scale**
  monday + Doctolib + Delivery Hero
- **19. Claude in Microsoft Foundry**
  Marlene Mhangami
- **20. What legal agents inherit**
  Jakob Emmerling, Legora
- **21. Claude on Google Cloud**
  Ivan Nardini, Schneider Larbi
- **23. Getting more out of Platform**
  Punit Shah

Claude Code

7 сессий — среда разработки и кейсы

- **02. What's new in Claude Code**
  Ralph Ramos
- **07. Beyond the basics**
  Daisy Hollman
- **08. Faster to production**
  Michael Cohen, Harrison Stall
- **09. From 1 to 80 at hypergrowth**
  Gabriel Grinberg, Yoav Orlev
- **16. AI-native engineering org**
  Fiona Fung
- **18. Signals that trade themselves**
  Tushara Fernando, regulated firm

Workshops

Hands-on с залом

- **10. Stop babysitting**
  Sid Bidasaria
- **14. Build a production agent**
  Michael Cohen
- **22. Proactive agent workflow**
  Maya Nielan

OPENING KEYNOTE

- **01. Six leaders, one stage**
  Ami Vora, Dianne Penn, Angela Jiang, Katelyn Lesse, Cat Wu, Boris Cherny

Сигнал направления: 13 из 24 сессий (54%) — must-watch для нетехнической бизнес-аудитории

Code w/ Claude в 2026 году. Не конференция для AI-разработчиков. Конференция для тех, кто принимает решения о применении AI в бизнесе

Источник: claude.com/code-with-claude/london (verified WebFetch 25.05.2026) — Under the Hood

Как читать эту карту

Каждая сессия в карте получила оценку по двум осям.

Track. Какой из четырёх тематических треков Anthropic относит сессию: Research (исследования и фреймворки), Claude Platform (платформенные функции и интеграции), Claude Code (среда разработки), Workshops (практические воркшопы с участием зала).

Релевантность для русскоязычного СМБ. - □ Must-watch. Прямая практическая ценность для владельца малого и среднего бизнеса. - □ Для технарей. Конкретная техника, требует базовых навыков работы с AI-стеком. - □ Контекст. Стратегический материал или нишевое применение.

Опенинг

▸ 01. Opening keynote

- **Спикеры:** Ami Vora, Dianne Penn, Angela Jiang, Katelyn Lesse, Cat Wu, Boris Cherny
- **Time:** 09:00-10:00, Main stage
- **Track:** General
- **Релевантность:** □
- **Главные анонсы:** Self-Hosted Sandboxes (public beta), MCP Tunnels (research preview), Claude Code Routines, +50% к лимитам Code, 20+ legal MCPs. См. Раздел 2.
- **URL:** Opening keynote

Утренний блок (10:30-12:30)

▸ 02. What’s new in Claude Code

- **Спикер:** Ralph Ramos
- **Time:** 10:30-11:00, Main stage

- **Track:** Claude Code
- **Релевантность:**
- **Суть:** Обзор всего нового в Claude Code за последние 3 месяца, включая Routines и удвоенные лимиты.
- **URL:** What’s new in Claude Code

03. Memory and dreaming for self-learning agents

- **Спикер:** Ravi Trivedi
- **Time:** 10:30-11:00, Breakout stage
- **Track:** Claude Platform
- **Релевантность:**
- **Суть:** Архитектура памяти и периодической переработки знаний (dreaming) для агентов с долгим горизонтом. Соответствует паттерну 5 из Раздела 3. См. Раздел 5, практика #3.
- **URL:** Memory and dreaming

04. Picking the right model

- **Спикер:** Lucas Smedley
- **Time:** 10:30-11:15, Workshop
- **Track:** Research
- **Релевантность:**
- **Суть:** Hands-on техники тестирования и сравнения моделей под конкретный use case. См. Раздел 5, практика #6.
- **URL:** Picking the right model

05. Coding is no longer the constraint: Scaling devex to teams and agents at Spotify

- **Спикер:** Niklas Gustavsson
- **Time:** 11:15-11:45, Main stage
- **Track:** Research
- **Релевантность:**
- **Суть:** Кейс Spotify по масштабированию developer experience через комбинацию команд людей и AI-агентов. См. Раздел 4, кейс Spotify.
- **URL:** Coding is no longer the constraint

▸ 06. Designing with Claude: From prompt to production

- **Спикер:** Dan Cary
- **Time:** 11:15-11:45, Breakout stage
- **Track:** Research
- **Релевантность:**
- **Суть:** Полный путь от промпта до продакшена с Claude: дизайн-процесс, метрики качества, переходы между фазами. См. Раздел 5, практика #7.
- **URL:** Designing with Claude

▸ 07. Beyond the basics with Claude Code

- **Спикер:** Daisy Hollman
- **Time:** 11:30-12:15, Workshop
- **Track:** Claude Code
- **Релевантность:**
- **Суть:** Продвинутые техники Claude Code: hooks, скиллы, MCP-серверы, кастомизация.
- **URL:** Beyond the basics with Claude Code

▸ 08. How to get to production faster with Claude Managed Agents

- **Спикеры:** Michael Cohen, Harrison Stall
- **Time:** 12:00-12:30, Main stage
- **Track:** Claude Code
- **Релевантность:**
- **Суть:** Managed Agents как путь сократить время от прототипа до production-готового агента. Связан с Self-Hosted Sandboxes. См. Раздел 2.
- **URL:** How to get to production faster

▸ 09. From one person to 80: Scaling a hypergrowth engineering org with Claude Code

- **Спикеры:** Gabriel Grinberg, Yoav Orlev
- **Time:** 12:00-12:30, Breakout stage
- **Track:** Claude Code
- **Релевантность:**

- **Суть:** Кейс роста инжиниринг-команды с 1 до 80 человек на базе Claude Code как основной рабочей среды. См. Раздел 4, кейс #1.
- **URL:** From one person to 80

Послеобеденный блок (12:30-15:00)

10. Stop babysitting your agents

- **Спикер:** Sid Bidasaria
- **Time:** 12:30-13:15, Workshop
- **Track:** Claude Code
- **Релевантность:**
- **Суть:** Воркшоп о том, как выбрать уровень автономии под уровень риска задачи. Контр-нарратив к мейнстриму «AI требует постоянного надзора». См. Раздел 6.
- **URL:** Stop babysitting your agents

11. AI with Claude on AWS: From Code to Orchestration

- **Спикер:** Antonio Rodriguez
- **Time:** 13:30-14:15, Workshop
- **Track:** Claude Platform
- **Релевантность:**
- **Суть:** Построение AI-агентов в AWS Bedrock с Claude. Для тех, кто на AWS-стеке.
- **URL:** AI with Claude on AWS

12. How Lovable vibecodes production software at scale

- **Спикер:** Fabian Hedin
- **Time:** 13:50-14:20, Main stage
- **Track:** Claude Platform
- **Релевантность:**
- **Суть:** Кейс Lovable: как платформа no-code AI-разработки выпускает production-софт в режиме vibecoding. Граничный пример: для большинства СМБ-задач этот режим избыточен, но для R&D полезен. См. Раздел 4, кейс Lovable.
- **URL:** How Lovable vibecodes production

▸ 13. The thinking lever

- **Спикер:** Alexander Bricken
- **Time:** 13:50-14:20, Breakout stage
- **Track:** Research
- **Релевантность:**
- **Суть:** Фреймворк о том, какую часть мышления оставить человеку, какую отдать AI, и где между ними точка опоры. См. Раздел 5, практика #8.
- **URL:** The thinking lever

Послеобеденный блок 2 (14:30-15:50)

▸ 14. Build a production-ready agent with Claude Managed Agents

- **Спикер:** Michael Cohen
- **Time:** 14:30-15:15, Workshop
- **Track:** General
- **Релевантность:**
- **Суть:** Воркшоп, продолжение сессии #08. Практическая часть с пошаговым построением production-готового агента.
- **URL:** Build a production-ready agent

▸ 15. Building AI-native at enterprise scale: monday.com, Doctolib, Delivery Hero

- **Спикеры:** Ruslan Semenov, Alex Kaluzny, Rodrigue Schäfer
- **Time:** 14:35-15:05, Main stage
- **Track:** Claude Platform
- **Релевантность:**
- **Суть:** Три компании из разных отраслей делятся опытом встраивания Claude. SaaS, медицина, доставка еды. Один общий паттерн. См. Раздел 4, кейсы #2-4.
- **URL:** Building AI-native at enterprise scale

▸ 16. Running an AI-native engineering org

- **Спикер:** Fiona Fung
- **Time:** 14:35-15:05, Breakout stage
- **Track:** Claude Code
- **Релевантность:**
- **Суть:** Управленческие практики для команд, где AI это часть рабочего процесса с первого дня. См. Раздел 5, organizational angle.
- **URL:** Running an AI-native engineering org

▸ 17. The capability curve

- **Спикер:** Jeremy Hadfield
- **Time:** 15:20-15:50, Main stage
- **Track:** Research
- **Релевантность:**
- **Суть:** Кривая возможностей AI-моделей: как меняется со временем то, что модели могут делать, и как калибровать ожидания под текущую фазу. См. Раздел 5 и Раздел 7.
- **URL:** The capability curve

▸ 18. Building signals that trade themselves: Running Claude Code inside a regulated investment firm

- **Спикер:** Tushara Fernando
- **Time:** 15:20-15:50, Breakout stage
- **Track:** Claude Code
- **Релевантность:**
- **Суть:** Кейс работы Claude Code внутри регулируемой инвестиционной компании. Узкий, но архитектурно полезный пример. См. Раздел 4, regulated industry бонус.
- **URL:** Building signals that trade themselves

Вечерний блок (15:30-17:35)

▸ 19. Build AI agents using Claude in Microsoft Foundry

- **Спикер:** Marlene Mhangami

- **Time:** 15:30-16:15, Workshop
- **Track:** Claude Platform
- **Релевантность:**
- **Суть:** Построение Claude-агентов внутри Microsoft Foundry (Azure AI). Для тех, кто на MS-стеке.
- **URL:** Build AI agents using Claude in Microsoft Foundry

20. What legal agents inherit from coding agents: Lessons from Legora

- **Спикер:** Jakob Emmerling
- **Time:** 16:05-16:35, Main stage
- **Track:** Claude Platform
- **Релевантность:**
- **Суть:** Кейс Legora: legal-агенты унаследовали архитектуру от code-агентов. Технический перенос паттернов, не маркетинг. См. Раздел 4, кейс #5.
- **URL:** What legal agents inherit from coding agents

21. Building with Claude on Google Cloud

- **Спикеры:** Ivan Nardini, Schneider Larbi
- **Time:** 16:05-16:35, Breakout stage
- **Track:** Claude Platform
- **Релевантность:**
- **Суть:** Интеграция Claude с Google Cloud (Vertex AI). Для тех, кто на Google-стеке.
- **URL:** Building with Claude on Google Cloud

22. Build a proactive agent workflow with Claude Code

- **Спикер:** Maya Nielan
- **Time:** 16:30-17:15, Workshop
- **Track:** Claude Code
- **Релевантность:**
- **Суть:** Воркшоп о построении проактивных workflows: агент сам инициирует действие на основе наблюдаемых сигналов, не ждёт команды. См. Раздел 5, практика #4.
- **URL:** Build a proactive agent workflow

23. Getting more out of the Claude Platform

- **Спикер:** Punit Shah
- **Time:** 16:50-17:20, Main stage
- **Track:** Claude Platform
- **Релевантность:**
- **Суть:** Консолидация платформенных функций: что есть, как сочетается, какие комбинации работают лучше всего.
- **URL:** Getting more out of the Claude Platform

24. The prompting playbook

- **Спикер:** Margot van Laar
- **Time:** 16:50-17:35, Breakout stage
- **Track:** Research
- **Релевантность:**
- **Суть:** Принципы промтинга для агентов, которые планируют, действуют, адаптируются. Завершающая сессия дня. См. Раздел 5, практика #5.
- **URL:** The prompting playbook

Итог

24 сессии разнесены по 4 трекам в следующих пропорциях.

- **Research:** 8 сессий (33%). Фундамент: фреймворки, методологии, кривые возможностей. Это то, что не устареет через 6 месяцев.
- **Claude Platform:** 9 сессий (37%). Платформенные функции и интеграции с тремя облаками (AWS, GCP, Azure) и с legal-вертикалью.
- **Claude Code:** 7 сессий (29%). Среда разработки и связанные кейсы.

Из 24 сессий **13 (54%)** получили оценку must-watch для русскоязычного СМБ. Это нетипично высокая концентрация для технической конференции, где обычно 70-80% материала остаётся в зоне «для технарей». Anthropic в Лондоне выстроил программу так, чтобы значительная часть сессий говорила не на языке инженеров, а на языке бизнеса.

Это сигнал направления. Anthropic осознанно расширяет аудиторию из разработческой среды в управленческую и предпринимательскую. Расчёт: продукт продаётся не разработчику, а покупателю с бюджетом, и значит говорить нужно с покупателем.

Code w/ Claude в 2026 году это конференция не для AI-разработчиков. Это конференция для тех, кто принимает решения о применении AI в бизнесе.

Запись каждой сессии будет опубликована Anthropic на claude.com/code-with-claude/london в течение 2-3 недель после события. Slot для recheck и обновления конкретных цифр из докладов: 02-09 июня 2026.

ROADMAP 18 МЕСЯЦЕВ

Что произойдёт в AI-индустрии к концу 2027 года.

Q3 2026

КОНЕЦ 2026

Регулируемые отрасли в США и ЕС

массово выбирают Anthropic как первого AI-партнёра

Q1 2027

+6-9 МЕСЯЦЕВ ДО РФ

Российский сегмент догоняет

Multi-vendor становится нормой для банков, страховых, медицины

Q2 2027

СЕРЕДИНА 2027

Методологический язык

6 runtime patterns / Stop Babysitting / Thinking Lever — стандарт индустрии

Q4 2027

КОНЕЦ 2027

Первые формальные регуляторные нормы

EU AI Act дополнения и US executive orders с терминами methodology layer

2028

К 2028 ГОДУ

Игра пяти уровней закончится

3-4 крупных AI-поставщика, каждый в собственной нише

К 2028

Anthropic

= регулируемый enterprise
+ методологический канон

К 2028

OpenAI

= массовый рынок
+ продуктовая экосистема

К 2028

Yandex · Sber

= суверенный рынок РФ
+ госконтракты

ВАША ТОЧКА ВХОЖДЕНИЯ

Окно 12 – 18 месяцев

Регулируемая отрасль: запускайте пилот на Anthropic-стеке с Self-Hosted Sandboxes и MCP Tunnels.
Массовый продукт: оставайтесь на OpenAI или Yandex, но выучите методологический язык.

В любом случае: встройте methodology layer в продукт до того, как индустрия зашумит.

Через год это будет очевидно. Сейчас это окно.

КОНЕЦ ВЫПУСКА 01

Это была только верхушка.

Под каждым из 8 разделов есть глубина, которая не вошла в PDF. Я разбираю её в своём канале Under the Hood — личная практика, не клуб, не курс.

ЧТО Я ПУБЛИКУЮ В КАНАЛЕ

01 Еженедельный разбор фронтир-сигналов AI-индустрии с расшифровкой стратегии

02 Архитектурные шаблоны под бизнес-задачи и российский комплаенс

03 Практические гайды по Claude Code, MCP, Cowork из реальной работы

04 Прямые ответы в комментариях, разбор ваших кейсов

TELEGRAM

@gorilla_under_hood

подписывайтесь и следите за разборами

АВТОР

Илья Утов

AI-практик, ex CTO Senses Media, Asigna