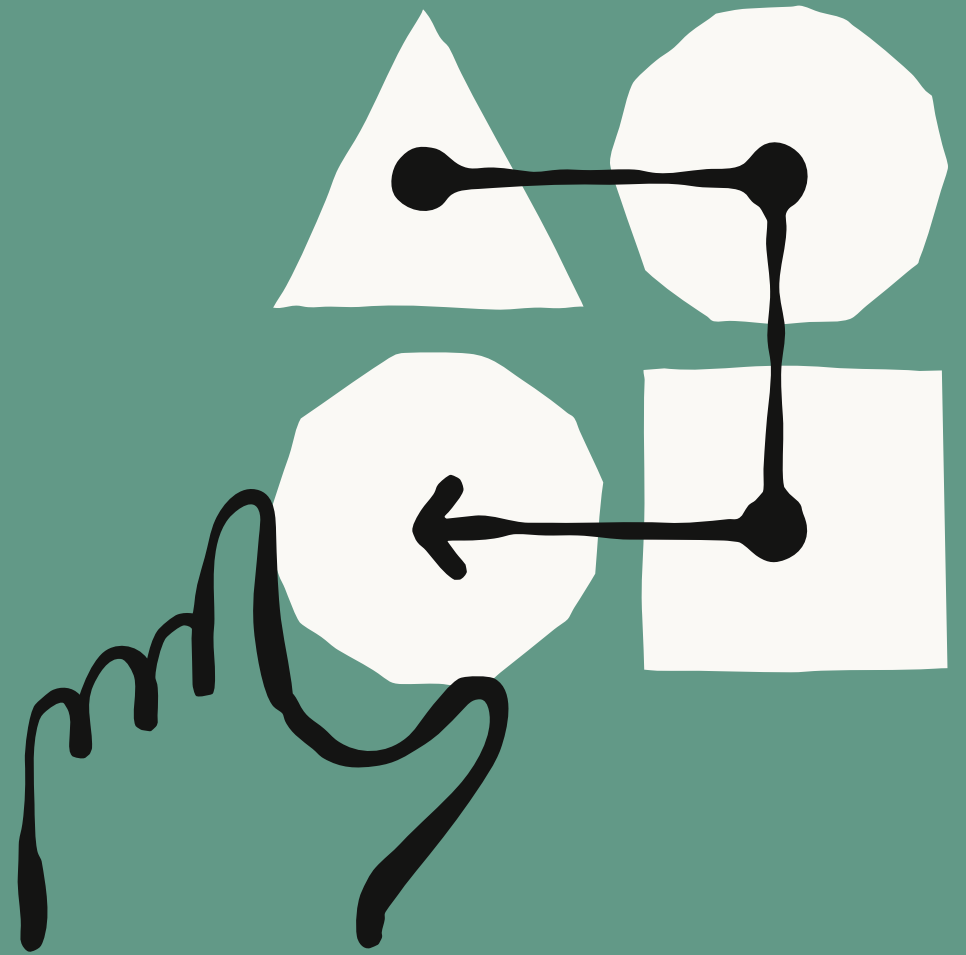


Руководство основателя: как построить ИИ- нативный стартап

Содержание

Жизненный цикл стартапа, перезагруженный для 2026 года	3
Как меняется роль основателя	5
Стадия идеи	8
Стадия MVP	15
Стадия запуска	21
Стадия масштабирования	25
Та же работа, новые правила	31
Ресурсы	33



Глава 1

Жизненный цикл стартапа, перезагруженный для 2026 года

Жизненный цикл стартапа, перезагруженный для 2026 года

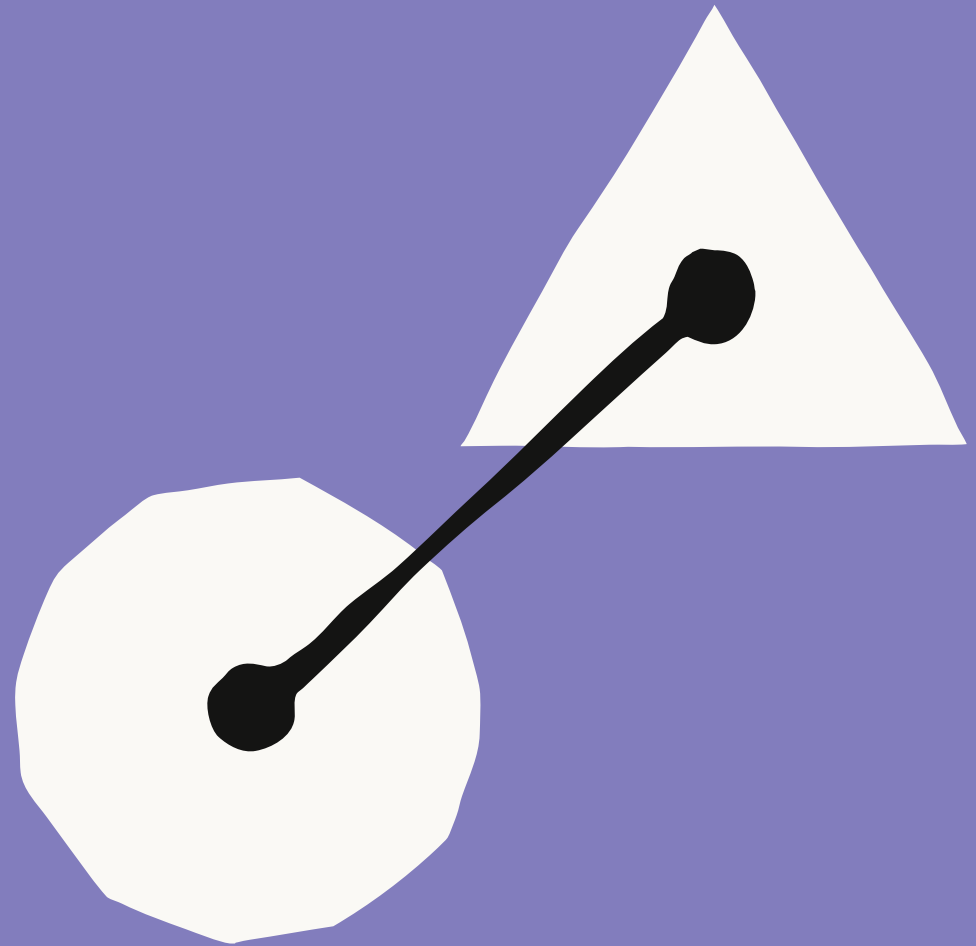
ИИ меняет то, как создаются стартапы. Основатели, которые никогда не писали ни строчки кода, сегодня выпускают полноценные продакшн-приложения, а компактный «единорог из десяти человек» превратился из истории дерзкого аутсайдера в осознанный план действий.

В 2026 году ИИ умеет писать продакшн-код, проводить исследования рынка, обобщать конкурентную среду, готовить материалы для инвесторов и автоматизировать операционные процессы. Устранив некогда крутую кривую обучения, с которой даже опытные технические основатели сталкивались при интеграции инструментов, платформ и систем, нужных для воплощения идеи, ИИ прежде всего уравнивал шансы в вопросе о том, кто вообще может запустить стартап или создать продукт.

В 2026 году хорошая идея продвигает основателя дальше, чем когда-либо. Агентное программирование сжимает то, что раньше требовало команды инженеров, до объёма работы, который основатель способен выполнить сам.

Традиционная траектория роста стартапа предполагает, что путь от идеи до масштаба выглядит так: проверить → привлечь инвестиции → нанять → построить → снова привлечь → вырасти → нанять ещё → повторить. Теперь ИИ стёр представление о том, что каждая новая фаза в жизненном цикле стартапа требует более крупной команды, иного набора навыков и нового раунда финансирования.

Это руководство заново описывает четыре ключевые стадии пути стартапа (Идея, MVP, Запуск и Масштабирование) с учётом новых реалий. Мы разбираем, как выглядит каждая стадия, когда ИИ лежит в основе вашего технического и организационного развития, какие инструменты подходят для каждой фазы и как основатели, использующие эти инструменты, сжимают сроки. Если вы готовы проложить кратчайший путь от идеи до выхода — читайте дальше.



Глава 2

Как меняется роль основателя

Как меняется роль основателя

Раньше основателей определяло то, что они умели делать: технические основатели писали код, а нетехнические вели операционную работу и закрывали сделки. Но модели, системы и ИИ-агенты, доступные основателям в 2026 году, разрушили стену между «теми, кто умеет строить» и «теми, у кого есть идеи, достойные воплощения».

ИИ-нативные стартапы коренным образом меняют то, что значит быть основателем. Теперь человек без инженерного опыта может создать продакшн-софт, воплощающий его идею, а технически подкованный основатель со скромными бизнес-знаниями легко подготовит стратегию выхода на рынок, финансовую модель и безусловно отполированный питч-дек.

Исторически основатели проводили большую часть времени в режиме исполнения: писали код, управляли людьми, разбирались с повседневной операционкой. В ИИ-нативном стартапе роль основателя гораздо меньше про индивидуальный вклад и гораздо больше про оркестрацию **агентов** — специализированных ИИ-помощников, которые умеют читать файлы, выполнять команды, запускать код и даже работать в вебе. Внимание основателя смещается выше по уровню — к работе высшего порядка: рождению идей и управлению системами (ИИ-агентами, инструментами и той небольшой командой, что есть), которые эти идеи воплощают.

Однако самый революционный результат ИИ как центральной инфраструктуры — это разблокировка нетехнических основателей, обладающих экспертизой в предметной области. Когда круг основателей расширяется за пределы людей с инженерным бэкграундом, появляются стартапы, построенные людьми с радикально разным жизненным опытом, — они решают реальные проблемы, которые традиционный конвейер технических основателей никогда не приоритизировал (а может, и не замечал).

Возможности ИИ-инструментов для компактных стартапов

Традиционная модель стартапа предполагала, что нужно нанимать инженеров, чтобы строить, продавцов, чтобы продавать, и операционных специалистов, чтобы вести бизнес. Численность штата считалась признаком организационного импульса и зрелости продукта.

Стартапы ранней стадии в 2026 году принципиально иные. Они предельно компактны по замыслу — нередко это один основатель или команда из нескольких человек. Выстраивая и техническое, и организационное развитие вокруг ИИ как инфраструктуры, они способны достичь подтверждения продукта, ранней выручки или даже прибыльности ещё до того, как начнут расширять команду. Есть три области, где ИИ особенно помогает стартапу работать как куда более крупная организация: исследования, агентное программирование и автоматизация процессов в ключевых бизнес-операциях.

Диалоговый интеллект и исследования

Думайте о нём как: дежурный эксперт в любой области

Подумайте обо всём, что основателю нужно знать в первый год и чего он почти наверняка не знает на старте: как настроить расчёт зарплат? Как планировать спринты разработки продукта? Как составить ёмкое инвесторское письмо?

У вопросов вроде этих на ранней стадии стартапа раньше был один и тот же ответ: «найди того, кто знает». Для основателя на бутстрэпе или предпосевной стадии это могло означать время, потраченное на сбор знаний вместо создания продукта, а то и сжигание части раннего капитала на консультанта. Теперь у них есть ИИ — дежурный эксперт в любой мыслимой области.

- **Глубокие исследования:** конкурентный анализ, оценка объёма рынка, финансовое моделирование
- **Подготовка документов:** питч-деки, кейсы, инвесторские письма, PRD
- **Партнёр по стратегическому мышлению:** анализ «адвоката дьявола», премо-темты, проработка сценариев, оптимизация дорожной карты

Агентное программирование

Думайте о нём как: инженер, который всегда доступен и которого ничто не блокирует

Раньше создание софта требовало технического сооснователя, подрядной студии разработки или достаточного запаса средств, чтобы нанять инженерную команду, прежде чем вы напишете хоть строчку продакшн-кода.

Агентные инструменты разработки теперь позволяют любому будущему основателю описать словами, что он хочет построить, и направлять ИИ генерировать, тестировать, отлаживать и рефакторить код продакшн-уровня со скоростью и в масштабе полноценной инженерной команды.

Путь от «у меня есть идея» до «у меня есть продукт» сжался. А роль основателя теперь сосредоточена на том, что строить и зачем, тогда как ИИ берёт на себя саму сборку реальной инфраструктуры, готовой к настоящим пользователям.

Автоматизация процессов

Думайте о ней как: операционная команда по требованию, работающая автоматически

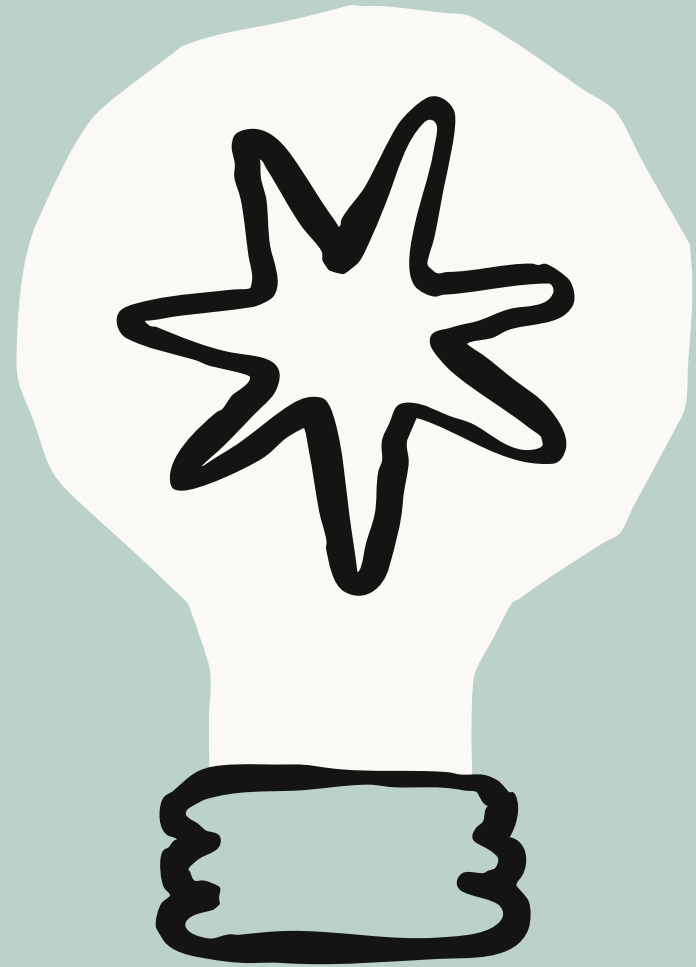
Даже когда основатель умеет исследовать как консультант и строить как инженерная команда, остаётся целый пласт работы за пределами стратегического планирования и разработки продукта, которую всё равно надо делать. Планирование встреч, обновление CRM, сбор еженедельных отчётов, поддержание документации в актуальном виде, публикация контента, отслеживание требований комплаенса, управление связующей тканью между инструментами и системами компании — всё это тоже должно происходить. В компактном стартапе эта нагрузка ложится в основном на основателя, и она серьёзно облагает налогом время и внимание, которые стоило бы направлять на решения более высокого порядка.

Автоматизация процессов с помощью ИИ-инструментов снимает этот налог. Повторяющиеся операционные задачи можно настроить так, чтобы они выполнялись автоматически: CRM обновляется при движении сделки, еженедельный отчёт собирается сам, а документация по продукту обновляется синхронно с изменениями продукта. И, что особенно важно, Claude Cowork интегрируется со взаимосвязанными системами, на которых работает стартап, — вашим инструментом управления проектами, коммуникационным стеком, источниками данных — без необходимости держать кого-то, кто будет строить и поддерживать эти интеграции. В стартапах «Дня ноль» этот кто-то почти всегда основатель.

Тайминг и оркестрация решают всё

Основатели, эффективно использующие исследовательские, автоматизационные и агентные возможности ИИ в программировании, могут построить стартап, работающий с куда большим рычагом, чем подсказывает численность их команды. Вдобавок они получают возможность посвящать большую часть своего времени и сил той работе, которая действительно важна.

Эта работа не идёт на автопилоте: основатель, оркеструющий ИИ-инструменты, должен знать, как (и когда) их применять. Остальная часть руководства посвящена целям и трудностям, с которыми основатели столкнутся на пути ИИ-нативного стартапа, и тому, как эффективно применять ИИ-инструменты на каждой стадии этого пути.



Глава 3

Стадия идеи

Стадия идеи

Каждый основатель стартапа начинает с одного и того же: с проблемы, о которой не может перестать думать. Это та фаза стартапа, где идея встречается с реальностью: успех стартапа в 2026 году требует дисциплины — не строить, пока доказательства этого не оправдывают.

Работа на этой стадии — это исследования, customer discovery, конкурентный анализ и честная оценка опровергающих доказательств, и всё это до того, как вы попросите Claude Code сгенерировать первую строку продакшн-кода.

Цель стадии идеи

На стадии идеи главная цель основателя — **исследовательская валидация**: сбор весомых доказательств того, что реальная проблема существует (и что предложенное вами решение действительно её устраняет), прежде чем вкладывать ресурсы в создание продукта.

Если говорить практически, стадия идеи — это череда вопросов, на которые основатель должен ответить примерно в таком порядке:

- Реальна ли эта проблема, конкретна ли она и достаточно ли часто встречается, чтобы строить вокруг неё?
- У кого именно она есть и является ли это рынком?
- Решает ли её кто-то ещё, и если да, то как и насколько хорошо?
- Что на самом деле должно уметь решение, чтобы устранить эту проблему, и делает ли это ваша идея?

Результаты этих вопросов складываются в ответ на один, главный вопрос: стоит ли это строить?

А это значит — стать конкретным, прежде чем двигаться. «Людам тяжело с отчётами о расходах» — это наблюдение. «Финансовые менеджеры в компаниях среднего размера тратят более четырёх часов в неделю на сверку заявок, потому что их нынешние инструменты не интегрируются с бухгалтерским ПО» — это проверяемая гипотеза.

Критерии выхода со стадии идеи

Условие выхода со стадии идеи — нахождение **соответствия проблемы и решения**. Вы собрали качественные доказательства, в первую очередь из реальных разговоров с людьми, что решаете настоящую проблему настоящих людей, ещё до того, как начнёте строить то, что её решает.

Вы готовы покинуть стадию идеи, когда можете утвердительно ответить на все три вопроса ниже:

1. **Реальна ли проблема и достаточно ли она конкретна?** Утвердительный ответ здесь требует, чтобы вы могли точно назвать, кто именно сталкивается с этой проблемой, как часто, насколько тяжело она на них сказывается и что они сейчас с ней делают.
2. **Решает ли ваше решение настоящую проблему?** Не ту проблему, которую вы предполагали изначально, а ту, которую вскрыл процесс валидации. Иногда это одно и то же, но далеко не всегда.
3. **Достаточно ли сигнала, чтобы оправдать стройку?** На этой стадии у вас никогда не будет полной уверенности, и ждать её — отдельная ловушка; но вам нужно достаточно качественных доказательств, чтобы переход к MVP был обоснованным решением, а не актом веры.

Трудности стадии идеи

Стадия идеи — это место, где совершается важнейшая работа всего пути стартапа, потому что именно здесь допускаются самые судьбоносные ошибки: ошибиться сейчас — значит быстро пустить молодое предприятие под откос. При этом большинство трудностей фазы идеи связаны с тем, что вы движетесь быстрее, чем оправдывает ваше понимание, поэтому основатели, действующие вдумчиво и обстоятельно, будут видеть устойчивый прогресс.

Принять стройку за валидацию

Трудность: Когда технические преграды сняты, увлечённый основатель рискует пропустить важнейшую работу на пути стартапа — проверку того, что его идея действительно является решением, которое людям нужно и которым они будут пользоваться.

Ещё до нынешней эпохи агентного программирования 42% стартапов проваливались из-за того, что строили то, что никому не было нужно. Теперь же агентные инструменты разработки вроде Claude Code резко сократили расстояние между «у меня есть идея» и «у меня есть продукт», и эта доля провалов будет только расти.

Хотя никогда ещё не было лучшего времени быть основателем с потрясающе хорошей идеей, скорость и лёгкость, с которой можно собрать прототип, похожий на продукт, парадоксальным образом несут в себе по-настоящему опасный экзистенциальный риск для ИИ-нативного стартапа.

Ещё совсем недавно стройка требовала реального времени разработчиков и бюджета, а собрать даже базовый прототип обычно занимало месяцы. Теперь, когда барьер технической разработки в основном исчез, ИИ делает слишком уж простым для основателя прыжок прямо в стройку без проверки полезности идеи в реальном мире.

Достижение соответствия проблемы и решения требует сначала подтвердить гипотезу, а потом строить, но многие начинающие (и даже опытные) основатели ошибочно полагают, что ИИ позволяет обойти это требование, превращая ход дела в «есть идея → сразу строю прототип → считаю само существование прототипа валидацией». Прототип становится поводом верить, что гипотеза была верна с самого начала, без всякой проверки того, так ли это на самом деле.

Рабочий прототип легко принять за конкретное доказательство того, что вы решаете настоящую проблему, но это не так. Ваш прототип служит другим — полезным инструментом проверки в разговорах с потенциальными пользователями. Именно эти разговоры и есть настоящее доказательство.

Преждевременное масштабирование

Трудность: Когда строить легко и мгновенно, можно масштабировать исполнение далеко впереди того, чего требует бизнес.

Преждевременное масштабирование означает, что вы привязываетесь к продуктовому пути ещё до того, как по-настоящему подтвердили, что этот путь стоит того, чтобы к нему привязываться.

Это всегда было убийцей стартапов, но ИИ сделал куда более лёгким незаметное падение в ловушку преждевременного масштабирования. Агентные ассистенты разработки настолько мощны, что легко уйти масштабированием исполнения далеко вперёд от подтверждения соответствия проблемы и решения, так и не приняв этого решения осознанно.

ИИ будет генерировать, тестировать, отлаживать и рефакторить код вокруг принципиально ошибочной посылки ровно с тем же энтузиазмом, с каким он берётся за отличную идею. Интеллект в этой системе — ваш. Главная заповедь на этой стадии — держать осмысление впереди стройки, особенно когда строить так быстро и так легко.

Потеря объективности

Трудность: Попросите ИИ-инструмент привести доказательства в пользу того, во что вы уже верите, — и он их найдёт. Подтверждающее искажение теперь идёт в комплекте с исследовательским движком.

Подтверждающее искажение всегда было профессиональным риском в стартапах: основатели по своей природе увлечены своими идеями. Теперь ИИ-инструменты дали этому искажению серьёзное усиление. Попросите ИИ подтвердить вашу идею стартапа — и он найдёт подтверждающие доказательства; попросите его оценить ваш потенциальный рынок — и он найдёт цифру, которая сделает ваш ТАМ привлекательным для финансирования.

ИИ следует вашему направлению, а это значит, что основатель, не задающий трудных вопросов, теперь может выстроить развёрнутое, выглядящее хорошо проработанным обоснование плохой идеи быстрее, чем когда-либо, при этом чувствуя полную уверенность, что он действительно проводит due diligence. Противоядие — тот же инструмент, лишь направленный в противоположную сторону: ИИ проверит идею на прочность так же тщательно, как и подтверждает её.

Когда исследования и структурированное состязательное мышление вскрывают доказательства того, что идея нуждается в пересмотре, — это сигнал к пивоту.

Как Claude помогает основателям на стадии идеи

Продвижение концепции вашего ИИ-нативного стартапа через стадию идеи может казаться бесконечным. Вы основатель, и вам просто хочется строить. Но эта важнейшая стартовая фаза по сути является упражнением в исследовании и валидации, а значит, нужно браться за инструменты, которые помогают мыслить строже, прежде чем с головой уйти в написание кода. Вот способы использовать **Claude** на всех его продуктовых поверхностях (Chat, Claude Cowork и Claude Code), чтобы пройти стадию идеи как можно быстрее, проводя при этом надлежащую проверку.

Chat, Claude Cowork или Claude Code: как выбрать подходящую поверхность Claude

ИИ облегчает основателям стартапов более быстрый выпуск, автоматизацию рутинных процессов и работу в масштабе, но поверхность, которую вы используете, имеет значение. Вот когда стоит выбирать Chat, Claude Cowork или Claude Code в зависимости от задачи.

Chat — для быстрых обменов, не покидая приложение, в котором вы уже находитесь. Используйте его для постоянных мелких задач управления компанией: вытащить ключевую мысль из плотного инвесторского письма, проверить утверждение перед заседанием совета директоров или разобраться в длинной переписке в Slack с вашей командой.

Claude Cowork — для интеллектуальной работы, которая реально занимает время: собрать данные из множества источников, осмыслить их и произвести что-то завершённое — документ, презентацию или таблицу. Подумайте о том, чтобы превратить папку с расшифровками клиентских звонков в тематический документ с выводами к следующему продуктовому ревью, собрать конкурентный обзор по десятку сайтов вендоров перед раундом или поставить регулярную задачу на утро понедельника, которая собирает метрики из подключённых инструментов и кладёт еженедельную сводку по KPI в общую папку.

Claude Code — это среда агентного программирования для инженеров вашей команды: прямой доступ к кодовой базе, Plan Mode, интеграция с git и локальные, IDE- или изолированные облачные среды. Именно здесь компактная команда выпускает функции в растущей кодовой базе, переносит унаследованный код со времён MVP и движется от прототипа к продакшену, не дожидаясь расширения штата.

Если задача такая...	Обратитесь к	Почему
Вопрос, переписывание, быстрый брейнсторм	Chat	Быстро, в формате диалога, без настройки
Исследование, анализ или готовый документ, собранный из ваших файлов и систем	Claude Cowork	Доступ к папкам, коннекторы, навыки, задачи по расписанию
Написание, тестирование или выпуск ПО	Claude Code	Доступ к кодовой базе, диффы, git, среды разработки

Под всеми тремя — один и тот же Claude; меняется лишь рабочее пространство вокруг него.

Определение и проверка на прочность гипотезы о проблеме

Ваша собственная экспертиза в предметной области и предварительные исследования уже породили гипотезу. Первая задача — заострить её до состояния, в котором она действительно проверяема. Claude особенно полезен здесь тем, что заставляет добиваться конкретики: у кого именно есть эта проблема, как часто, насколько тяжело и что эти люди делают с ней сейчас? Формулировка проблемы, которая не может точно ответить на эти вопросы, не готова к валидации.

- **Упражнение:** Поработайте с Claude, чтобы заострить формулировку проблемы до проверяемой гипотезы. Например, «*проверка договоров занимает слишком много времени*» — это не осмысленно проверяемое утверждение. А вот «*внутренние юридические команды в компаниях среднего размера тратят больше трёх дней на цикл проверки одного договора, потому что правки ведутся в почтовых цепочках, а не в едином документе с контролем версий*» — вполне проверяемо.

Ваш следующий шаг — попросить Claude выступить против вашей идеи и найти опровергающие доказательства, которые refute вашу гипотезу. Это может вскрыть негативные рыночные сигналы, провалившихся конкурентов, паттерны поведения клиентов и структурные препятствия, которые поддакивающий синтез тихо отодвинул бы на задний план.

Цель — подойти к customer discovery, уже проверив свои допущения на прочность против сильнейших доступных контраргументов, чтобы ознакомительные интервью с пользователями были по-настоящему открытыми, а не поиском подтверждения.

Заметка: Использование Claude в роли структурированного «адвоката дьявола» — это базовый сценарий на каждой стадии жизненного цикла ИИ-стартапа.

Исследование рынка и картирование конкурентной среды

Оценка ваших конкурентов

Существует специфичный для стартапов феномен — пренебрежение конкурентами: склонность настолько сосредоточиться на собственном видении и исполнении, что вы систематически недооцениваете то, что делают другие в том же пространстве. К счастью, ИИ предлагает противоядие: попросите Claude привести самый убедительный аргумент в пользу того, почему конкурент в этом пространстве решений преуспеет, а вы — нет.

Claude может разобрать, почему их подход на самом деле лучше, почему клиенты выберут их, почему ваши потенциальные отличия могут оказаться не такими защищёнными, как вы думаете.

прямые конкуренты, косвенные конкуренты, потенциальные покупатели и смежные игроки, которые могут зайти в ваше пространство. Затем попросите его аргументировать, почему каждый уровень несёт реальную угрозу вашему успеху, — а не ту версию угрозы, которую проще всего отмахнуть.

Исследование рынка

Claude Code может синтезировать общедоступные отзывы клиентов, чтобы вскрыть повторяющиеся жалобы и неудовлетворённые потребности. Бонус: по сути это бесплатное качественное исследование клиентов ваших конкурентов.

- **Упражнение:** Поручите Claude Cowork синтезировать отзывы о конкурентах по вашим ключевым источникам и определить главные жалобы, которые существующие решения так и не устранили. Если ваша гипотеза устраняет одну или несколько из них — это весомое доказательство соответствия проблемы и решения. Если нет — это тоже стоит знать.

Claude Cowork также может извлекать нужные сведения и цифры из плотных отраслевых отчётов, аналитических документов и материалов исследований рынка; затем эти очищенные, обобщённые входные данные становятся идеальным контекстом для аналитической работы Claude.

- **Упражнение:** Постройте модели TAM/SAM/SOM на основе общедоступных данных и проверьте на прочность стоящие за ними допущения. Определите, расширится рынок, консолидируется или зрел; этот контекст влияет на то, как вы думаете о тайминге и дифференциации. Составьте карту покупателей: у кого бюджет, кто влияет на решения и один ли это человек.

Анализ трендов

Наконец, используйте Claude, чтобы улавливать ранние индикаторы, говорящие вам, входите ли вы в рынок в подходящий момент. Отслеживайте сабреддиты и группы LinkedIn, где уже идут разговоры о вашей проблеме, и точные формулировки, которыми пользователи описывают свои затруднения. Попросите Claude найти аналогичные рынки, где похожая проблема была решена, и извлечь, что сработало, а что нет. Вскройте регуляторные, технологические или демографические тренды, которые могут ускорить возможность или поставить её под угрозу.

технологические или демографические, — которые могут существенно повлиять на ваш рынок в ближайшие два года, и оценить, является ли каждый из них попутным ветром или встречным для вашей конкретной гипотезы.

Заметка: Исследование рынка и картирование конкурентов в этом разделе — не разовое упражнение. Вы продолжите делать открытия и развивать своё мышление в течение стадий MVP и Запуска, поэтому важно повторять эти упражнения всякий раз, когда ваша гипотеза эволюционирует.

Планирование и проектирование customer discovery

Качество того, что вы узнаете, общаясь с потенциальными пользователями продукта, определяется (1) качеством вопросов, которые вы задаёте, и (2) тем, задаёте ли вы их правильным людям. Claude особенно полезен для проведения customer discovery — включая то, с кем говорить, что спрашивать и как осмыслять услышанное.

С кем говорить

Точный профиль целевой аудитории бесконечно ценнее длинного списка контактов — включая конкретные должности, типы компаний, структуры команд и уровни старшинства тех, кто с наибольшей вероятностью остро ощущает проблему. Отсюда определите, где этих людей реально можно достать — сообщества, мероприятия, группы LinkedIn и рабочие пространства Slack, где они собираются, — и постройте систему приоритизации того, к кому обращаться первыми, исходя из того, насколько близко они к проблеме.

Что спрашивать

Определив целевую аудиторию, используйте Claude, чтобы построить сам каркас интервью: правильные вопросы, в правильном порядке, выстроенные так, чтобы вскрывать, что люди действительно делают, а не что, по их мнению, они бы сделали. Типичная ошибка начинающего основателя — задавать общий вопрос с открытым концом о будущем («вы бы пользовались чем-то таким?») вместо конкретного расспроса о релевантном прошлом («расскажите о последнем случае, когда вы столкнулись с этой проблемой»).

Claude может отметить, где ваши черновые вопросы подталкивают респондента, слишком широки или иным образом склонны порождать шум вместо сигнала. Claude также поможет вам в проектировании уточняющих вопросов, чтобы прощупать уклончивые ответы или докопаться до сути расплывчатых ответов на важные вопросы.

Если ваша гипотеза затрагивает более одной персоны, Claude может также спроектировать разные наборы вопросов для каждой. У финансового менеджера и финансового директора разные отношения с одной и той же проблемой, а единый каркас интервью эту разницу сгладит.

- **Упражнение:** Сначала составьте вопросы для интервью вручную, затем попросите Claude провести их аудит. Попросите его конкретно отметить любой вопрос, который является наводящим, обращённым в будущее, слишком широким или склонным порождать социально желаемый ответ, а не честный. Затем попросите предложить уточняющий вопрос для двух-трёх моментов интервью, где уклончивость наиболее вероятна.

Анализ после интервью

После каждого разговора используйте Claude для разбора: дайте ему свои заметки и попросите определить, что подтвердило вашу гипотезу, что ей бросило вызов и что было

по-настоящему неожиданным. Собрав пакет интервью, прогоните весь свой набор заметок через Claude Cowork, чтобы вскрыть повторяющиеся темы, противоречия и сильнейшие сигналы в обоих направлениях. Затем верните этот обобщённый результат Claude и попросите его отметить, где ваше собственное прочтение данных может подгоняться под желаемое, а не под то, что есть на самом деле.

заметки и составить два списка: доказательства, поддерживающие вашу гипотезу, и доказательства, которые ей противоречат. Если первый список значительно длиннее второго, спросите Claude, отражает ли эта асимметрия то, что реально есть в данных, — или то, что вы надеялись найти.

Привлечение клиентов и планирование встреч

Используйте Claude Cowork, чтобы автоматизировать операционную нагрузку вокруг составления списка контактов, проведения рассылок и планирования пользовательских интервью.

Claude Cowork может использовать профиль целевой аудитории, который вы определили вместе с Claude (включая должности, типы компаний и уровни старшинства), чтобы исследовать и собрать структурированный список потенциальных собеседников с проверенной контактной информацией. Затем он составляет персонализированные письма для рассылки в масштабе, подстраивая каждое под роль и контекст конкретного человека.

По мере поступления ответов он подключается к Gmail и Google Календарю через MCP, чтобы вести переписку, обрабатывать запросы на встречи и ставить интервью в календарь. Рабочий процесс продолжается: Claude Cowork генерирует черновики дополнительных писем с заданной периодичностью (например, напоминание на седьмой день для тех, кто не ответил) и обновляет вашу таблицу отслеживания по мере завершения каждого шага, так что вы всегда знаете, на каком этапе воронки находится каждый собеседник.

- **Упражнение:** Дайте Claude Cowork ваш подтверждённый профиль целевой аудитории интервью и попросите его собрать список потенциальных собеседников, составить персонализированную последовательность писем и подготовить таблицу отслеживания со столбцами для статуса рассылки, периодичности напоминаний и завершённости интервью. Затем дайте ему вести координацию, пока вы сосредоточитесь на подготовке к самим разговорам.

Спроектируйте итоговую концепцию решения

Вы проделали работу по валидации: проблема реальна, вы знаете, у кого она есть, и у вас есть концепция решения, которую поддерживают доказательства. Используйте Claude, чтобы развивать и оспаривать вашу концепцию решения со всех сторон: где пробелы? Какие есть альтернативы? Что должно быть истинным, чтобы это решение работало в масштабе? Это важная контрольная точка реальности: действительно ли этот дизайн устраняет проблему, которую вскрыл процесс валидации, а не ту проблему, которую вы изначально предполагали на старте?

- **Упражнение:** Представьте свою концепцию решения Claude и попросите его выявить три допущения, на которые ваш дизайн опирается сильнее всего. Затем спросите, что должно быть истинным, чтобы каждое допущение выполнялось, и каковы последствия, если хотя бы одно из них не выполнится.

Постройте лёгкий прототип с Claude Code

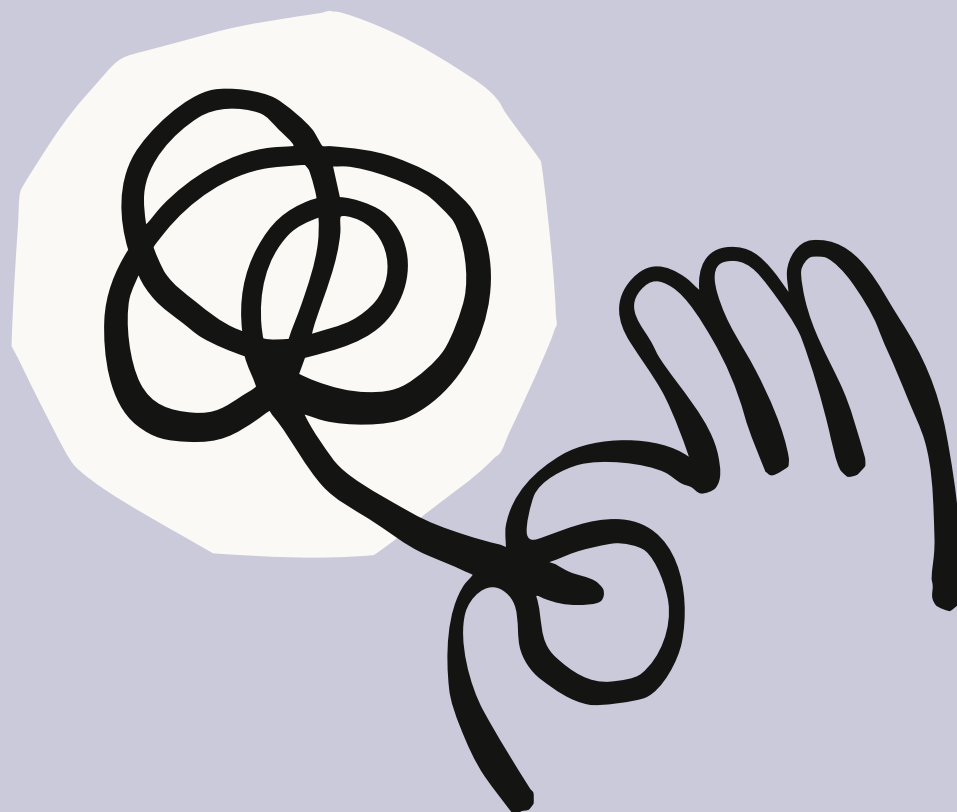
Теперь самое интересное: с подтверждённой гипотезой и проверенной на прочность концепцией решения вы наконец готовы построить что-то.

Это тот момент стадии идеи, когда в кадр входит Claude Code. Даже если вы всё это время что-то мастерили, именно сейчас наступает точка, в которой вы создаёте свой официальный лёгкий прототип: минимальную поверхность, нужную, чтобы показать вашу идею живому человеку и получить настоящую реакцию.

Вы строите не реальный продукт (пока), а собираете работающий образец вашей идеи, чтобы использовать его в разговорах с клиентами и инвесторами. Реальные пользователи, реагирующие на то, что можно по-настоящему потрогать, расскажут вам то, чего не смогли бы и десяток discovery-интервью о проблеме и решении. Раньше вы устанавливали, что проблема, которую вы решаете, реальна; теперь вы просите потенциальных пользователей взаимодействовать с предложенным решением.

- **Упражнение:** Определите единственное ключевое взаимодействие, на котором держится ваше решение. Поручите Claude Code построить только его. Когда оно готово, покажите его пяти людям из вашего подтверждённого профиля целевой аудитории и попросите попробовать. То, что вы узнаете в этих пяти разговорах, и определит, продолжать ли строить или вернуться к чертёжной доске.

Достичь конца стадии идеи — это огромный скачок вперёд в гонке ИИ-стартапов, потому что теперь вы делаете ставку не на догадку, а исполняете на основе доказательств. Дальше идёт стадия MVP, где направляющий вопрос основателя меняется с «стоит ли это строить?» на «что именно нам стоит построить в первую очередь?», а основная роль ИИ смещается с исследовательского партнёра на строительную бригаду.



Глава 4

Стадия MVP

Стадия MVP

Многие основатели относятся к стадии MVP как к фазе строительства, но стадия MVP по-прежнему по сути является упражнением в сборе доказательств. Разница в том, что теперь вы собираете доказательства о решении, а не о проблемном пространстве; в частности — о том, считает ли реальная, идентифицируемая группа людей продукт достаточно ценным, чтобы пользоваться им, возвращаться к нему, платить за него и/или рассказывать о нём другим.

Цели стадии MVP

Как основатель ИИ-нативного стартапа, ваша цель — **превратить подтверждённую проблему в работающий продукт**, которым реальные пользователи действительно будут пользоваться. Это не полная версия со всеми функциями из дорожной карты, а наименьшая, максимально сфокусированная итерация вашей идеи, которая показывает реальное решение реальным пользователям и порождает реальное доказательство product-market fit.

В то же время то, как вы строите сейчас, определяет, что будет возможно потом. Это значит, что у стадии MVP есть вторая, не менее важная цель — **двигаться быстро, не накапливая такой технический долг**, который нарастает сложным процентом и настигнет вас в тот момент, когда реальные пользователи появятся в заметном количестве.

И наконец, **вложение в постоянный контекст** с первого дня — это то, что удерживает ИИ в роли мультипликатора силы, а не источника энтропии. В ИИ-нативном стартапе ваша кодовая база — это то, над чем вы сотрудничаете с ИИ сессия за сессией, и это делает её читаемость фундаментальной. Основатели, пропускающие спецификации, архитектурные решения и контекстные файлы (вроде CLAUDE.md), упираются в предсказуемую стену, где каждая новая сессия требует заново объяснять кодовую базу, а сгенерированные ИИ изменения дрейфуют от изначального замысла.

Критерии выхода со стадии MVP

Условие выхода со стадии MVP — подлинное доказательство product-market fit: подтверждение того, что конкретная, идентифицируемая группа пользователей сочла продукт достаточно ценным, чтобы возвращаться к нему (удержание), платить за него (выручка) или рассказывать о нём другим (рекомендации).

Трудности стадии MVP

На стадии MVP главные заповеди основателя — скорость и здравое суждение. Трудности здесь сосредоточены на том, способны ли вы построить правильную вещь, правильным способом, достаточно быстро, чтобы это имело значение, не срезая углы, которые потом дорого вам обойдутся.

Агентный технический долг

Трудность: Поскольку ИИ по сути убирает все естественные узкие места, которые когда-то контролировали то, что попадает в продакшен, скорость гарантирована. Но когда скорость — единственная переменная, которую основатели учитывают при сборке MVP, они рискуют накопить технический долг, расплатиться с которым будет тяжело.

Некоторый технический долг уместен на стадии MVP — при понимании, что им нужно управлять до масштабирования. Он накапливается постепенно и может быть погашен со временем или в отдельном спринте. Однако ИИ-технический долг нарастает сложным процентом.

Без спецификаций и архитектурных ограничений, записанных где-то, где ИИ может их прочитать, каждая сессия заново выводит фундаментальные решения с нуля, и эти решения дрейфуют. В итоге вы получаете кодовую базу, за которой нет связной ментальной модели, — не потому, что какой-то отдельный кусок плох, а потому, что куски никогда не проектировались так, чтобы стыковаться. Это реальная проблема, и она, как правило, всплывает поздно.

Принять ложный product-market fit за настоящий

Трудность: ИИ-инструменты могут порождать впечатляющие ранние цифры, но это не гарантия того, что рынку нужен ваш продукт.

Ранний импульс — одно из самых психологически мощных переживаний, доступных основателю. После недель или месяцев работы по валидации и аккуратной, дисциплинированной стройки выпуск продукта ощущается как подтверждение того, что вы были правы всё это время.

Агентные инструменты разработки могут помочь вам достичь этого момента быстрее, чем когда-либо, но ранняя тяга — это не то же самое, что product-market fit. Энергия запуска порождается эфемерными силами — друзьями основателя, потенциальными покупателями из других портфельных компаний вашего инвестора или заголовком на Hacker News, который даёт всплеск. К сожалению, ничто из этого надёжно не предсказывает, что произойдёт на шестой или двенадцатой неделе, когда первоначальный толчок угаснет.

Расползание границ без трения

Трудность: Когда строить кажется легко и почти бесплатно, всегда найдётся ещё одна крутая функция, которую хочется добавить, или ещё один краевой случай, который хочется обработать. Это расползание границ может принести больше вреда, чем пользы.

Расползание границ всегда было риском для стартапа. Разница теперь в том, что традиционный сдерживающий механизм — реальная стоимость инженерного времени — больше не работает так же, когда добавление функции занимает полдня вместо спринта.

Трудность здесь в том, что каждое отдельное добавление обосновано. Конечно, продукт должен обрабатывать этот краевой случай; конечно, пользователи захотят этот сценарий. В моменте это не ощущается как расползание границ, потому что каждое требует так мало усилий при агентном программировании, — но по мере того, как продукт разрастается за пределы своих изначальных границ, вы рискуете потерять направление и импульс.

Противоядие — письменное определение границ, созданное до начала стройки, описывающее, что продукт делает, чего он намеренно не делает и какое конкретное доказательство от реальных пользователей оправдало бы добавление чего-то нового. Это смещает точку принятия решения с «стоит ли нам это строить?» на «достаточная ли масса пользователей сказала нам, что не может получить ценность от продукта без этого?».

Уязвимость от неопытности

Трудность: Основатели, использующие ИИ-инструменты, чтобы поскорее выкатить приложение на рынок, не разобравшись сперва в фундаментальных принципах безопасности, в итоге подвергают своих пользователей предотвратимым рискам.

Жёсткая правда в том, что агентные инструменты разработки генерируют код, который работает, а не код, который по своей сути безопасен. Рабочий код — это легко, потому что либо функция работает, либо нет. Уязвимости в безопасности невидимы, пока их не проэксплуатируют, а значит, нет естественной обратной связи, которая предупредила бы начинающего основателя, что что-то не так. Однако выпуск живого MVP реальным пользователям означает реальные данные, реальную уязвимость и реальные последствия, если что-то пойдёт не так.

Пренебрежение безопасностью не является чем-то совсем новым для ИИ-нативных проектов. Стартапы на бутстрэпе во все эпохи часто откладывали вопросы безопасности до позднего этапа стройки, иногда — до самого порога продакшен-запуска. Ревью безопасности до того, как любой пользователь прикоснётся к вашему приложению или решению, — это минимальный ответственный порог для выпуска минимально жизнеспособного продукта в мир.

Как Claude помогает основателям на стадии MVP

Определите архитектуру до того, как начнёте строить

Прежде чем Claude Code напишет хоть строку продакшен-кода, используйте Claude, чтобы определить и задокументировать архитектурные решения, которые будут управлять всем, что строится на этой стадии: какие паттерны соблюдать, каких зависимостей избегать, какие компромиссы делаются и почему. Этот результат послужит сфокусированным документом архитектурного контекста и задаст ограждения, внутри которых будет работать Claude Code.

Без этого контекста каждая сессия начинается с нуля, и Claude Code вынужден выводить собственные структурные допущения. Если позволить Claude Code строить без ограждений, получится кодовая база, которая будет работоспособной, но структурно несвязной, а итерировать и масштабировать несвязные кодовые базы — в конечном счёте трата времени и токенов. Рано или поздно наступает точка, где код неизбежно рушится, заставляя вас перестраивать всё с нуля.

- **Упражнение:** Прежде чем открыть Claude Code, откройте Claude и опишите, что вы строите: ключевую проблему, которую это решает, пользователей, которым служит, и масштаб, которого вы реально ожидаете в ближайшие полгода. Попросите его помочь определить архитектурные принципы, которые должны управлять сборкой вашего MVP, зависимости, которых стоит избегать с учётом ваших ограничений, и компромиссы, на которые вы осознанно идёте на этой стадии.

Затем ~~сохраните этот результат как файл(ы) CLAUDE.md в формате markdown~~. Это ваш документ архитектурного контекста: первый артефакт вашей стройки и тот, от которого зависит каждая последующая сессия. Файлы CLAUDE.md служат инструкциями уровня проекта для Claude Code, предоставляя специфичный для проекта контекст и инструкции, которые автоматически считываются Agent SDK при запуске в директории. Функционально это постоянная «память» вашего проекта.

Определите и соблюдайте границы вашего MVP

Расползание границ без трения — один из определяющих режимов провала MVP эпохи ИИ. Так же, как вы определили и задокументировали архитектуру приложения, вам нужно определить и границы вашего MVP до того, как будет построена хоть одна функция.

Claude может помочь вам создать документ о границах, описывающий, что ваш MVP-продукт делает, чего он намеренно не делает, и критерии добавления функций: какое конкретное доказательство от реальных пользователей оправдало бы добавление чего-то нового на данном этапе.

Когда всплывают идеи новых функций — а они непременно всплывут, — используйте Claude, чтобы проверить на прочность, настоящий ли это сигнал от пользователей или энтузиазм основателя, наряженный под продуктивное мышление.

Постройте ваш MVP с Claude Code

Когда архитектура и границы определены, Claude Code становится основным инструментом сборки MVP. Используйте его, чтобы генерировать, тестировать, отлаживать и итерировать вашу кодовую базу, но относитесь к каждой сессии как к исполнению продуктовых решений, которые вы уже приняли, а не как к поводу подбросить новые.

Начинайте каждую сессию Claude Code с того, чтобы (1) пересмотреть документ о границах и (2) предоставить модели ваш документ архитектурного контекста CLAUDE.md. Завершайте каждую сессию его обновлением — любыми решениями, которые сессия вскрыла. Цель — кодовая база, структуру которой вы можете объяснить, а не просто кодовая база, которая запускается.

- **Упражнение:** Создайте простой шаблон сессии для вашей работы с Claude Code, включающий документ архитектурного контекста, конкретную задачу этой сессии и любые ограничения или паттерны, которые нужно соблюдать. В конце каждой сессии добавляйте краткую запись в журнал контекстного документа с описанием того, что было построено, какие решения приняты и какие допущения сессия внесла. Пять минут документирования на сессию — это дешёвая страховка от архитектурного дрейфа, который перерастает в неуправляемую кодовую базу.

Ревью безопасности до того, как к продукту прикоснётся хоть один пользователь

Как основатель ИИ-нативного стартапа, ваша обязанность — знать, что находится в вашей кодовой базе, понимать любые потенциальные векторы уязвимости и не выпускать очевидные уязвимости реальным пользователям, которые доверяют вам свои данные.

Claude может провести полезное первичное ревью безопасности сгенерированного ИИ кода и помочь выявить распространённые уязвимости. Это хорошая привычка — встроить его в процесс перед выпуском. Однако это не замена инструментам безопасности, а в более ответственных случаях — и человеку-ревьюеру, и основатели, которые относятся к нему как к замене, как раз и оказываются героями историй об утечках.

Claude Code Security идёт дальше: он сканирует кодовые базы на уязвимости и предлагает точечные патчи для проверки человеком, вскрывая проблемы, которые традиционные методы могут упустить.

Заметка: На момент публикации этой электронной книги Claude Code Security находится в ограниченном бета-доступе, поэтому проверьте текущую доступность, прежде чем встраивать его в свой рабочий процесс.

- **Упражнение:** Прежде чем разворачивать что-либо для реальных пользователей, прогоните код вашего ключевого приложения через Claude с конкретным заданием: проверить аутентификацию и обработку сессий, раскрытие данных в ответах API, валидацию ввода и риски инъекций, а также зависимости с известными уязвимостями. Относитесь к каждой находке серьёзно и оцените, требует ли она исправления, с проверкой человеком всего, что касается аутентификации, секретов или работы с данными.

Постройте систему измерений до запуска

Основатели, которые ошибочно принимают раннюю тягу за product-market fit, — обычно те же, кто начал отслеживать данные после запуска, выбирая метрики так, чтобы оценить, что работает, а не вскрыть, что не работает. Противоядие — выстроить систему измерений до того, как появится первый пользователь.

Используйте Claude, чтобы определить, какие метрики важны для вашего конкретного продукта, каковы ориентиры и какие паттерны в данных составляют подлинный product-market fit в отличие от лестного шума. В частности: задайте ориентиры удержания, критерии активации и цели для Дня 7 и Дня 30 до выпуска вашего MVP.

Затем определите, как выглядит ложноположительный результат для вашего конкретного продукта: регистрации без активации, выручка без удержания или первоначальный энтузиазм без повторного использования, например. Когда данные придут, попросите Claude построить состязательный довод против вашей собственной тяги: что сказал бы скептик об этих цифрах?

Управляйте логистикой discovery и обратной связи пользователей

Как только реальные пользователи появляются в продукте, операционный слой быстро расширяется. Claude Cowork берёт на себя важную, но рутинную работу: создание и поддержание списков контактов пользователей, проведение рассылок, планирование сессий обратной связи, разбор баг-репортов и отслеживание циклов итераций. Те же интеграции MCP, что управляли логистикой discovery на стадии идеи, применимы и здесь.

Держите человека в цикле сбора для тонкого разбора обратной связи пользователей. Пользователь, говорящий, например, «это здорово, но я бы хотел, чтобы оно ещё...», требует интерпретации: это ключевая потребность или приятное дополнение? Это специфично для этого клиента или характерно для сегмента? Недостающая функция и есть настоящая проблема — или что-то выше по течению, в онбординге? Ни один инструмент не ответит на эти вопросы.

- **Упражнение:** Настройте Claude Cowork вести ваш цикл обратной связи стадии MVP: составлять рассылки для вашего списка ранних пользователей, планировать сессии обратной связи, проектировать структурированный процесс приёма баг-репортов и запросов функций и писать еженедельную сводку того, что поступило. Сначала просматривайте сводку сами; после этого можно попросить Claude проанализировать информацию, чтобы выловить значимые моменты, которые вы могли упустить.

Итерируйте к доказательству, а не к завершённости

Стадия MVP заканчивается, когда у вас есть подлинное доказательство product-market fit, каким бы «завершённым» ни ощущался продукт. Заявить, что вы достигли product-market fit и теперь готовы перейти с фазы MVP на стадию запуска, — это в конечном счёте упражнение в суждении, сочетающее интуицию основателя с собранными доказательствами. Однако есть несколько полезных лакмусовых тестов:

- **Тест Шона Эллиса:** Спросите своих активных пользователей: «Что бы вы почувствовали, если бы больше не могли пользоваться этим продуктом?» Если более 40% отвечают «очень расстроился бы» — это значимый индикатор PMF.
- **Тест усилия:** До product-market fit удержание требует постоянного вмешательства — частых контактов, поощрений, личных напоминаний и героической энергии основателя, расходуемой на то, чтобы удерживать вовлечённость пользователей. После product-market fit продукт начинает делать эту работу сам. Когда всё начинает тянуть, а не толкать, этот сдвиг в усилиях — один из самых ясных сигналов того, что произошло что-то настоящее.

В конечном счёте ни одна отдельная точка данных не подтверждает product-market fit, потому что это паттерн, который должен удерживаться на протяжении нескольких циклов итераций, прежде чем вы сможете окончательно его констатировать.

Делайте пивот, когда этого требуют доказательства

Что, если даже после всей этой работы вы никак не можете прийти к product-market fit? То, что ваши результаты не подтверждают направление, с которого вы начинали, — это не провал, это работающая система: стадия MVP и задумана для того, чтобы вскрыть эту информацию до того, как вы переинвестируете в неверный ответ.

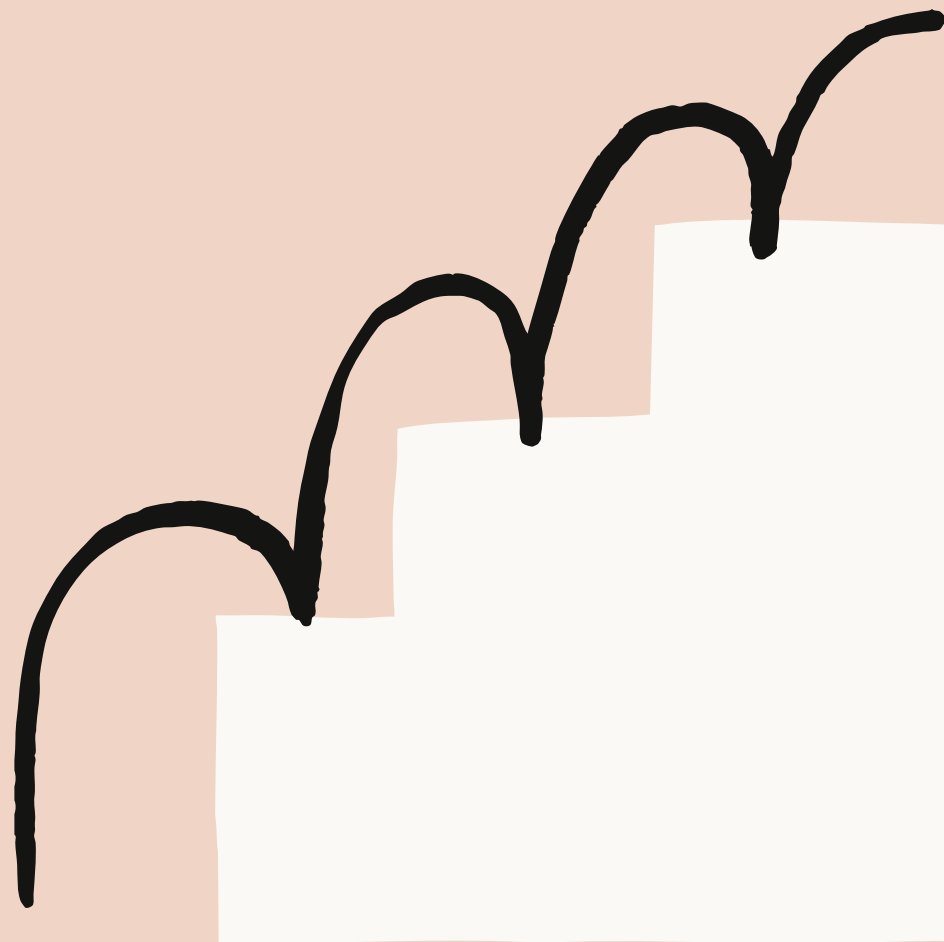
Когда данные не поддерживают ваш текущий продукт, используйте Claude, чтобы проработать, о чём эти данные вам говорят.

- **Исследование альтернативных сегментов клиентов.** Возможно, пользователи, которые не конвертируются, изначально не были правильной целевой аудиторией. Часто нужная аудитория уже есть в ваших данных, просто недооценена.
- **Корректировка ценностного предложения вашего продукта.** Возможно, у вас правильная аудитория, но ваш MVP просто не находит у пользователей отклика. Корректировка онбординга, формулировок или акцента на ключевых функциях способна это исправить без изменения того, что вы построили.

Будьте открыты к тому, что разрыв может оказаться достаточно глубоким, чтобы потребовать более фундаментального изменения.

- **Упражнение:** Если вы прошли три или более цикла итераций без значимого продвижения к вашим ориентирам product-market fit, используйте Claude, чтобы провести диагностику, прежде чем решать, что делать дальше. Дайте ему ваши данные по удержанию, обратную связь пользователей и вашу изначальную гипотезу о проблеме и задайте три вопроса:
 - Есть ли в этих данных сегмент, который реагирует иначе, чем остальные?
 - Разрыв между задуманной ценностью и переживаемой ценностью — это проблема позиционирования или проблема продукта?
 - Что должно быть истинным, чтобы текущий продукт нашёл подлинный PMF, и реалистичен ли такой сценарий с учётом того, что вы наблюдаете?

Пусть ответы определяют, корректируете ли вы, делаете пивот или возвращаетесь на стадию идеи.



Глава 5

Стадия запуска

Стадия запуска

Если стадия MVP была о том, чтобы доказать, что ваш продукт заслуживает существовать, то стадия запуска — о том, чтобы доказать, что ваш бизнес заслуживает расти.

Цели стадии запуска

На стадии запуска основатели стартапов должны **превратить раннюю тягу в воспроизводимый, устойчивый двигатель роста**. Помимо того чтобы довести продукт до готовности к продакшену, вы также должны укрепить инфраструктуру под ним, одновременно строя вокруг продукта настоящую компанию.

Стартапы естественно сосредоточены на основателе на стадиях идеи и MVP, потому что вам нужна полная ситуационная осведомлённость и тесные циклы обратной связи. Теперь же основатели, которые всё ещё пытаются лично держать каждую нить, становятся узким местом стадии запуска. Цель не в том, чтобы убрать себя из компании, а в том, чтобы **построить операционные системы, освобождающие ваше внимание** для решений, которые может принять только основатель.

Критерии выхода со стадии запуска

Условия выхода со стадии запуска три составляющих:

1. **Рост воспроизводим и опирается на каналы.** Вы не просто удерживаете пользователей — вы приобретаете их предсказуемо через конкретные каналы с понятной юнит-экономикой: CAC, LTV и срок окупаемости — это цифры, которые вы знаете и можете отстаивать.
2. **Продукт выдерживает продакшен-нагрузки.** Инфраструктура укреплена, безопасность и комплаенс в порядке, а надёжность держится в реальных продакшен-условиях (а не только в тех, что вы тестировали).

3. **Операции работают без узких мест в лице основателя.** Процессы существуют, автоматизация на месте. Вы больше не тот человек, который лично занимается поддержкой, разбором обращений, планированием спринтов или отчётностью.

Трудности стадии запуска

Нахождение product-market fit — самая трудная задача раннего жизненного цикла стартапа. Теперь трудностью основателя становится его удержание. Стадия запуска — это место, где компании, нашедшие реальную продуктовую тягу, всё ещё могут развалиться, если организация, окружающая и поддерживающая продукт, не поспекает. Вот режимы провала, за которыми стоит следить.

Технический долг приходит к оплате

Трудность: Кодовая база MVP, построенная ради скорости и валидации, работала достаточно хорошо, чтобы доказать работоспособность продукта, но продакшен-трафик, новые функции и растущая сложность теперь вскрывают срезанные углы.

На стадии MVP накопить некоторый технический долг было разумным компромиссом ради скорости. На фазе запуска этот долг начинает накапливать проценты, и чем дольше им не занимаются, тем дороже его исправлять.

Решение состоит из систематического архитектурного аудита для выявления структурных слабостей, точечного рефакторинга для устранения худших из них и заметного расширения покрытия тестами, чтобы следующий раунд работы над функциями не вернул те же проблемы.

Основатель становится узким местом

Трудность: На стадии MVP присутствие основателя в каждом цикле было активом. На стадии запуска, по мере роста объёма поддержки, накопления продуктовых решений и умножения операционной сложности, тот же инстинкт становится ограничением.

Переход от выполнения работы к проектированию систем, которые её выполняют, — один из самых трудных сдвигов в жизненном цикле стартапа. Поскольку редко бывает чёткий момент, когда он происходит, риск — пропустить его вовсе и остаться в режиме строителя, пока организация вокруг вас стопорится. Тревожные признаки того, что это происходит: решения, которые должны занимать час, теперь требуют недели, прежде чем у вас дойдут до них руки; запросы в поддержку копятя, потому что ответ знаете только вы; а операционные задачи случаются только тогда, когда вы лично о них вспоминаете.

Средство — тотальный аудит всего, чем вы занимаетесь лично, от мельчайшей задачи до решений с наивысшими ставками, чтобы определить, что можно систематизировать, что можно делегировать, а что действительно по-прежнему заслуживает времени и внимания основателя.

Безопасность и комплаенс больше нельзя откладывать

Трудность: Держать меры безопасности и комплаенса простыми было нормально для MVP, но теперь, с реальными пользователями, реальными данными и потенциально корпоративными контрактами на столе, это становится уязвимостью.

На стадии MVP, с горсткой бета-пользователей и без чувствительных данных в продакшене, уязвимости безопасности были теоретическими рисками. Однако гипотетическое становится вполне реальным риском в тот момент, когда ваш продукт выходит в продакшен с реальными пользователями, которые на него полагаются. Более того, требования комплаенса, не применимые к прототипу, определённо применимы в тот момент, когда вы обрабатываете клиентские данные, проводите платежи или продаёте в регулируемые отрасли.

Средство — систематическое ревью безопасности и комплаенса до прихода продакшен-масштаба, а не после; и относитесь ко всему, что всплывёт, как к обязательному устранению, а не рекомендации, прежде чем придёт следующая волна пользователей.

Расширение до того, как вы готовы

Трудность: Новые рынки и возможности финансирования выглядят как возможности роста. Они же могут стать местом, где product-market fit умирает.

Первоначальная тяга, которую вы построили, реальна, но она ещё и специфична для вашей ранней аудитории. Слишком раннее расширение в рынок, который заметно отличается от исходного, привносит новые модели поведения пользователей, требования комплаенса, платёжную инфраструктуру и базовые ожидания, под которые ваш продукт не проектировался. Внезапно появляется слишком много новых переменных, и вы теряете способность ясно интерпретировать собственные данные. Вы также рискуете запустить свою исходную базу пользователей в погоне за новой и непроверенной аудиторией.

Как Claude помогает основателям на стадии запуска

Все три формы Claude в полном ходу на стадии запуска, и они поддерживают друг друга: каждый инструмент производит результаты, которые становятся входными данными для двух других. Результаты органично складываются сложным процентом, и основатель, использующий все три инструмента вместе, получает больше, чем сумму их частей.

Именно это делает структурно возможной модель сверхкомпактного стартапа. Когда Claude Code строит продукт, Claude Cowork строит вокруг него компанию, а Claude помогает операционализировать этот продукт и организационные знания, маленькая команда может работать как компания во много раз большего размера.

Устраните технический долг до того, как он нарастёт

Ваша кодовая база MVP работает, но ей также нужен систематический проход по устранению — в поисках любого технического долга, который может стать структурной уязвимостью.

Сначала используйте Claude Code, чтобы провести полный архитектурный аудит: выявить, где кодовая база хрупка, какие срезанные углы станут дорогими в поддержке и где покрытие тестами достаточно тонкое, чтобы следующий раунд работы над функциями вернул те же проблемы.

Передайте находки аудита Claude Code обратно в Claude, чтобы расставить приоритеты и упорядочить работу по устранению: что нужно исправить до следующего релиза, что может подождать спринт, а что представляет собой приемлемый текущий долг с учётом вашей нынешней стадии. Это также момент задокументировать архитектурные решения, принятые на стадии MVP (те, что жили в вашей голове, потому что записать их было некогда). Перенос их в CLAUDE.md сейчас гарантирует, что каждая будущая сессия Claude Code начнётся с общего понимания того, как и почему была спроектирована система.

- **Упражнение:** Поручите Claude Code провести аудит вашей кодовой базы MVP и составить приоритизированный список структурных слабостей, пробелов в покрытии тестами и кандидатов на рефакторинг. Затем передайте этот список Claude и попросите упорядочить работу по устранению по нескольким спринтам: значимые проблемы, которые нужно решить в первую очередь, то, что можно вести параллельно с разработкой функций, и то, что может подождать.

Постройте системы, заменяющие внимание основателя

Построение операционных систем, освобождающих ваше внимание для обязанностей, которые может взять только основатель, требует точного знания того, куда уходит ваше внимание. Используйте Claude Cowork, чтобы провести структурированный аудит вашей текущей операционной нагрузки, документируя каждую повторяющуюся задачу, каждое решение, которое ложится на ваш стол, и каждый рабочий процесс, который случается только потому, что вы лично о нём помните. Затем поручите Claude Cowork разложить этот реестр по категориям: что можно полностью автоматизировать, что требует человека, но не обязательно вас, и что действительно требует суждения основателя.

Когда аудит завершён, используйте Claude Cowork, чтобы спроектировать логику рабочих процессов для кандидатов на автоматизацию: что запускает каждый процесс, каковы правила принятия решений, как выглядит результат и куда он уходит по завершении.

Сделайте безопасность и комплаенс отдельным продуктовым рабочим потоком

Используйте Claude Code, чтобы вскрыть проблемы на уровне кода, которые часто всплывают в аудитах SOC 2, GDPR или HIPAA, и стандарты, которые требует ваш целевой рынок. Это вскроет и уязвимости, и пробелы комплаенса. Передайте эти находки Claude, чтобы он помог расставить приоритеты в работе по устранению и спроектировать контроли, аудит,

журналирование и управление доступом, которые корпоративные покупатели запросят, прежде чем подпишут. **Заметка:** Сканирование с помощью ИИ — это подспорье, но не замена квалифицированному ревью комплаенса.

Далеестройте рабочий поток комплаенса в ваш цикл разработки, а не ведите его как разовый проект; документация по комплаенсу нуждается в постоянном поддержании и обновлении. Для основателей, подступающих к корпоративным контрактам или международным рынкам, это также момент, когда сканирование безопасности Claude Code может помочь вам подготовиться к независимой оценке безопасности.

- **Упражнение:** Проведите ревью безопасности на уровне кода с Claude Code, ориентированное на фреймворки, которые требует ваш целевой рынок. Передайте результат Claude и попросите его произвести две вещи: приоритизированную последовательность устранения проблем безопасности и список документации и контролей, которые вам нужно будет подготовить, чтобы пройти ревью комплаенса от потенциального корпоративного покупателя.

Выстройте процессы продуктового менеджмента, которые вы пропускали

Стадия запуска требует набора лёгких, воспроизводимых процессов, которые могут работать без необходимости вмешательства основателя, чтобы запуститься или функционировать. Используйте Claude, чтобы спроектировать, как будут выстроены ваш продуктовый таймлайн и рабочие циклы, что должна включать спецификация, прежде чем Claude Code прикоснётся к функции, как баг-репорты разбираются и маршрутизируются и что охватывает ваш еженедельный отчёт по метрикам и как он распространяется.

Когда проектирование процессов завершено, используйте Claude Cowork, чтобы построить и вести операционный слой: планирование спринтовых церемоний, маршрутизация входящих баг-репортов в нужное место, сбор еженедельных метрик из подключённых источников данных и поддержание цикла обратной связи, который держит пользовательские сигналы поступающими в продуктовые решения.

- **Упражнение:** Попросите Claude спроектировать лёгкую операционную систему продуктового менеджмента: заданный ритм спринтов, шаблон минимальной спецификации, дерево решений для разбора багов и еженедельную сводку метрик, собираемую из ваших реальных источников данных. Затем настройте Claude Cowork внедрить и вести повторяющиеся операционные элементы этой системы — планирование, маршрутизацию и сборку отчётов — так, чтобы они выполнялись по расписанию без вашего участия.



Глава 6

Стадия масштабирования

Стадия масштабирования

На фазе масштабирования роль основателя заново перецентрируется со строителя на публичного руководителя. Продукт по-прежнему в центре, но ваша личная повседневная работа всё больше становится про саму компанию. Ваше внимание должно расширяться на новые активности стадии масштабирования — брифинги для аналитиков и IPO-роуд-шоу, — даже когда вы стремитесь сохранить компактное, ИИ-центричное структурное преимущество.

Цели стадии масштабирования

Работа по масштабированию технической инфраструктуры продолжается и теперь дополняется работой по масштабированию самой организации и её взрослению в бизнес.

На стадии масштабирования вы смотрите на переход от тысяч пользователей к миллионам и от одного рынка ко многим. На всех предыдущих стадиях рост был чем-то, что можно было нащупать, оставаясь близко к пользователям и корректируя курс по данным из тесных циклов обратной связи плюс здоровая доля интуиции основателя. Теперь же цель — **построить систематический рост**, поддерживаемый зрелыми организационными операциями.

Для ИИ-нативного стартапа вашей целью должно быть **построить защитный ров через накопленную глубину** — основанную на экспертизе, которую вы встроили в продукт, на глубине интеграции вашего продукта с другими инструментами и платформами, на которые полагаются ваши пользователи, и на проприетарных системных данных и рабочих процессах. Основатели, которые последовательно строили в одном направлении, на единой инфраструктуре, теперь обладают чем-то по-настоящему трудным для воспроизведения.

На этой стадии публичные инвесторы, аналитики, регуляторы, корпоративные отделы закупок и покупатели применяют больше давления — наряду с бóльшим скепсисом, — потому что ставки теперь выше. **Ваш продукт и организация должны выдерживать внешнее изучение:** не только возможности того, что вы построили, но и

управление, позиция по комплаенсу, финансовые контроли и стратегический нарратив, которые его окружают.

Критерии выхода со стадии масштабирования

Условие выхода на стадии масштабирования — уже не отдельная веха, а пороговое событие: компания устойчива, даже когда основатель всё больше не управляет повседневными операциями напрямую. Вы продемонстрировали систематический рост; построили организационное управление и инфраструктуру комплаенса, удовлетворяющую самых требовательных внешних ревьюеров; и у вас есть твёрдый ответ на вопрос: «Если хорошо профинансированный игрок скопирует ваш продукт сегодня, останутся ли ваши пользователи?»

На практике этот порог обычно принимает одну из трёх форм: устойчивая прибыльность в масштабе, который больше не требует внешнего капитала, готовность к IPO или поглощение. Все три требуют, чтобы ваш рост был систематическим и поддающимся аудиту, ваш продуктовый ров выдерживал пристальное изучение, а ваша организация была операционно зрелой и устойчивой.

Когда это так, можно поздравить: ваш стартап превратился из ставки в бизнес.

Трудности стадии масштабирования

Делегирование операционного слоя

Трудность: Операционные системы стадии масштабирования должны работать надёжно и устойчиво, без того чтобы за ними нянчились. Для основателя, который был погружён в дела с первого дня, этот переход может оказаться не меньшим психологическим вызовом, чем структурным.

Ваша работа на стадии запуска состояла в создании систем; на фазе масштабирования она становится (1) доведением этих систем до полной надёжности и (2) затем — реальным доверием к ним.

Это труднее, чем кажется. Даже если вы основатель, который хорошо делегирует, не всегда очевидно, что передать, а что оставить на своей тарелке. Передадите слишком много, слишком быстро — особенно ИИ-автоматизированным системам — и критические решения будут приниматься без важного контекста, который может дать только основатель. Удержите слишком долго — и вы рискуете стать узким местом.

Фундаментальная трудность здесь — выявить институциональные знания, которые живут только в голове основателя или в недокументированных процессах, и затем кодифицировать их в системы, которые задокументированы, поддаются аудиту и передаваемы.

Масштабирование технических операций

Трудность: Клиенты больше не оценивают только ваш продукт; они хотят знать, что ваша организация может быть надёжным инфраструктурным партнёром.

Технические трудности на первых трёх стадиях стартапа были сосредоточены на кодовой базе: построить правильное решение, не накапливая технический долг, а затем укрепить безопасность и комплаенс для реальных пользователей. Достигнув фазы масштабирования, вы сталкиваетесь с трудностью во всём, что построено вокруг кодовой базы: в создании инфраструктуры поддержки, документации и гарантий надёжности, которые сигнализируют о зрелости.

Более крупные клиенты и институциональные покупатели, подписывающие многолетние контракты, хотят видеть всё это, прежде чем подпишут, и будут спрашивать с вас за это после того, как подпишут. Однако та же ИИ-инфраструктура, что довела вас досюда, помогает вам построить выделенные функции поддержки с заданным временем отклика и документацией, которой инженерная команда нового клиента действительно сможет пользоваться.

Масштабирование организационных функций

Трудность: Компании на стадии масштабирования обычно нужна организационная инфраструктура — наём, расчёт зарплат, бухгалтерия и юридические операции — независимо от того, сколько людей ею управляют.

На стадии запуска систематизация операций означала автоматизацию рабочих процессов, поглощавших внимание основателя. Стартапу на стадии масштабирования теперь нужно вырастить ещё более широкий и в некотором смысле более ответственный набор операционных функций — финансовую отчётность, мониторинг комплаенса, управление контрактами и поддержку клиентов, и это лишь некоторые.

Построение GTM-функции

Трудность: У органического роста есть потолок, и большинство основателей на стадии масштабирования упираются в него ещё до того, как им вообще приходилось строить настоящую функцию выхода на рынок.

Рост на стадиях идеи, MVP и запуска часто берёт начало в продажах, которые ведёт сам основатель, — от удачно подгаданного поста на Product Hunt до личных отношений с ранними клиентами. Однако такой органический рост работает лишь до определённого предела, и большинство стартапов упираются в него на фазе масштабирования. Признаки — выполаживающиеся кривые пользователей, растущая стоимость привлечения клиентов и воронка, которая движется, только когда в неё лично вовлечён основатель.

Рост на стадии масштабирования требует построения отдельного двигателя роста, чтобы достучаться до новых и более широких аудиторий вашего продукта. Однако большинство основателей стартапов, вероятно, никогда не вели такие вещи, как маркетинг, продажи и программы отношений с аналитиками. Полноценное GTM-движение требует не только выстраивания новых систем и процессов, но и создания голоса бренда и истории о том, как вы хотите говорить о вашем продукте. Потому что на этой стадии жизненного цикла стартапа он вам понадобится, чтобы достучаться не только до отдельных новых пользователей, но и до целых целевых аудиторий — инвесторов и корпоративных покупателей.

К счастью, GTM-функция не обязана быть большой, чтобы быть эффективной, и та же ИИ-инфраструктура, что построила продукт, может загрузить-«забугтрапить» его вывод на рынок.

Как Claude помогает основателям на стадии масштабирования

Ранние стадии стартапа используют Claude как фундаментальную инфраструктуру самого продукта: исследовательский партнёр для валидации идеи, инженерная команда, которая проектирует и строит прототип, и ИИ-операционный слой, делающий возможным стартап одного основателя. Основатели ИИ-нативных стартапов, дошедшие до стадии масштабирования,

могут теперь использовать Claude, Claude Code и Claude Cowork, чтобы продолжать масштабироваться тем же способом, каким они строили.

Передача повседневных задач Claude Cowork

Начинайте стадию масштабирования с трезвого взгляда на то, куда вам сейчас важнее всего вкладывать своё время и внимание, — что может быть трудностью для начинающих основателей, которые никогда раньше не строили бизнес. Claude поможет, составив список того, чем на этой стадии должны заниматься только вы, — это могут быть решения о нарративе продукта, отношения с советом директоров, корпоративные сделки и разговоры основателя с основателем. Всё, чего нет в этом списке, — кандидат на делегирование или автоматизацию через Claude Cowork.

- **Упражнение:** Используйте Claude, чтобы составить карту узких мест вашего текущего операционного слоя: каждый рабочий процесс, решение и согласование, которые сейчас проходят через вас. Затем попросите Claude экстраполировать, что произойдёт с каждым из них, когда вы будете недоступны неделю. Процессы, которые застопорятся, — это те, где вы всё ещё достаточно погружены, чтобы тормозить продвижение.

Как они соотносятся с реестром приоритетов и обязанностей основателя, который вы составили вместе с Claude?

Дальше пришло время проверить на прочность, что системы, которые вы уже построили, действительно готовы масштабироваться вместе с вашим бизнесом по мере его роста.

- **Упражнение:** Используйте Claude, чтобы составить карту ваших текущих рабочих процессов, а затем спросите его, что произойдёт с каждым из них, когда вы будете недоступны неделю. Процессы, которые застопорятся, — это те, где критерии передачи, пути эскалации или обработка исключений всё ещё нуждаются в доработке. Claude поможет проанализировать точки отказа и порекомендует подходящие исправления, чтобы вы могли по необходимости обновить или заменить автоматизации Claude Cowork.

Масштабируйте технические операции в инфраструктуру корпоративного уровня

По мере масштабирования покупателям нужны заверения в том, что вашему продукту и вашей организации можно доверять как долгосрочной инфраструктуре. Техническая работа по-прежнему идёт внутри кодовой базы, как и всегда, но теперь появляется техническая работа и вокруг кодовой базы.

Первый шаг — превратить институциональные знания в систему, которая масштабируется. Используйте Claude, чтобы составить и поддерживать письменную инфраструктуру, которую ожидают увидеть корпоративные закупки, — документацию по продукту, плейбуки поддержки и SLA.

Параллельно поручите Claude Code провести аудит и укрепить кодовую базу под конкретные стандарты надёжности и безопасности, которых требуют корпоративные контракты, и выстроить техническую инфраструктуру поддержки, которую поддержке на базе Discord никогда не приходилось обеспечивать: журналирование, мониторинг, инструменты реагирования на инциденты и слой наблюдаемости, который делает SLA реально исполнимыми.

Затем Claude Cowork ведёт сам операционный слой корпоративной поддержки: маршрутизацию тикетов, процессы эскалации, обновления документации, запускаемые изменениями продукта, отслеживание продлений и ритмы отчётности, на которые опирается корпоративный customer success. Вместе эти трое дают маленькой команде поддержку уровня куда более крупной организации — именно то, что подписание многолетнего корпоративного контракта требует от вас продемонстрировать.

- **Упражнение:** Выберите трёх ваших самых требовательных потенциальных клиентов или определите трёх идеальных клиентов для вашего продукта, которых вы мечтаете подписать. Попросите Claude провести анализ пробелов: какую документацию, SLA и инфраструктуру поддержки корпоративный отдел закупок в каждой из этих компаний ожидал бы увидеть перед подписанием многолетнего контракта и где вы сейчас недотягиваете? Используйте результат, чтобы упорядочить техническую и документационную работу между Claude Code и Claude Cowork.

Постройте настоящую GTM-функцию

Хватка основателя довела вас досюда, но масштабирование вашего стартапа требует создания и внедрения настоящей стратегии выхода на рынок. ИИ поможет вам построить, а затем и вести этот полноценный GTM-двигатель.

Claude может помочь со строительством фундаментальных GTM-ресурсов с нуля: сегментация рынка, архитектура сообщений, стратегия отношений с аналитиками, плейбуки продаж и нарративы метрик для инвесторов, которые становятся важны, как только вы начинаете говорить с публичными инвесторами, корпоративными покупателями и аналитиками с Уолл-стрит. У каждой из этих аудиторий свой словарь, и она оценивает вас по своим

стандартам; задача Claude — перевести ценностные предложения вашего продукта в подход продуктового маркетинга, релевантный для каждого сегмента аудитории.

Теперь Claude Cowork может стать вашим слоем тактического исполнения: контент-конвейеры, исходящие последовательности писем, логистика брифингов для аналитиков, ритмы ньюсрума и PR, гигиена CRM, отчётность по воронке и множество повторяющихся циклов, которые превращают GTM-стратегию в реальное коммерческое движение.

Там, где GTM-движение требует инфраструктуры продуктового маркетинга — интерактивных демо-сред, документации по интеграциям, песочниц-тенантов, справочников по API, технических одностраничников, — Claude Code может построить её для вас. Покупатели ожидают возможности оценить ваш продукт технически, и на фазе масштабирования видео в Loom и презентация для продаж уже не годятся. Это также инфраструктура, которая позволяет вашему GTM-движению работать асинхронно: хорошо построенная демо-среда закрывает сделки, пока вы на заседаниях совета директоров.

Превращение экспертизы в предметной области и институциональных знаний в контекст для ИИ

Многие основатели сверхкомпактных стартапов строят узкоспециализированные приложения или инструменты для реальной проблемы, которую они переживают или наблюдают из первых рук в конкретной отрасли. Агентный ИИ теперь даёт возможность основателям, которые никогда не писали ни строчки кода, использовать свою экспертизу в предметной области, чтобы строить продукты, решающие сложные задачи. Claude, Claude Code и Claude Cowork — каждый играет свою роль в превращении знаний основателя в нарастающую продуктовую специфичность.

Использование Claude, чтобы захватывать, упорядочивать и оттачивать знания основателя, помещает экспертизу в предметной области туда, куда продукт может дотянуться. Через продолжительные диалоги, проекты и память основатель может поделиться всем, что знает, — отраслевым жаргоном, регуляторными подводными камнями, краевыми случаями, затруднениями, причинами, по которым очевидные ответы на эту проблему не работают, — в структурированный, доступный для поиска контекст. **Навыки** затем могут кодифицировать повторяющиеся рабочие процессы (например, «как я провожу аудит коммерческой аренды», «как я разбираю форму приёма пациента») в переиспользуемые рутины, которые Claude выполняет одинаково каждый раз. За месяцы это становится проприетарным субстратом знаний, с которым не сравнится ни один универсальный ИИ.

Вынесение ваших знаний в предметной области наружу с помощью Claude становится бесценным для кодирования отраслевых краевых случаев в ваш продукт: универсальный ИИ

инструмент медицинского биллинга ломается, например, на заявках по программе лекарств 340B, а у вашего есть для них специальная логика. Claude Code помогает вам перевести типичные затруднения, с которыми сталкиваются другие профессионалы в вашей области, в логику валидации, уточнения промптов или интеграцию через MCP с нишевой отраслевой системой, о которой ваши конкуренты и не слышали. В результате глубина и широта вашего приложения или инструмента непрерывно нарастают сложным процентом так, как конкуренты просто не могут воспроизвести.

- **Упражнение:** Определите один краевой случай, который универсальный конкурент в вашей вертикали наверняка обработал бы неверно. Поработайте с Claude Code, чтобы построить для него выделенный тест-кейс (не юнит-тест) на основе сценария, который вы реально наблюдали. Каждый раз, когда всплывает похожий краевой случай, добавляйте его. Ваш набор тестов становится картой вашего защитного рва.

Превратите накопленные данные пользователей в защищённое преимущество

По мере того как пользователи взаимодействуют с вашим продуктом, они генерируют поведенческие сигналы (то есть какие результаты они принимают, а какие отвергают), которые информируют дорожную карту продукта. Со временем вы узнаете конкретные паттерны, предпочтения и краевые случаи вашей собственной базы пользователей. Именно это мы имеем в виду под нарастающей сложным процентом ценностью: каждое улучшение делает продукт полезнее, что ведёт к большему использованию, что порождает больше обратной связи, что ведёт к новым улучшениям.

Эти данные привязаны ко времени, контекстно-специфичны и невозможны для воссоздания подражателем: вы попросту не можете купить поведенческий отпечаток тысяч пользователей, которые оттачивали свои рабочие процессы внутри вашего продукта.

Claude может помочь провести аудит любых собранных вами данных о взаимодействии пользователей, выявить в них поведенческие паттерны с наивысшим сигналом и спроектировать цикл обратной связи, который превращает текущее использование в систематическое улучшение модели.

- **Упражнение:** Дайте Claude сводку данных о взаимодействии с вашим продуктом: что вы собирали, как долго вы это собирали и что вы знаете о том, как пользователи взаимодействуют с вашим продуктом со временем. Попросите его выявить три поведенческих паттерна с наивысшим сигналом в этих данных и спроектировать цикл обратной связи, который превращает каждый из них в систематическое улучшение модели. Затем попросите помочь вам составить одностраничный нарратив о защитном рве для продуктового маркетинга: историю того, как работает ваш маховик данных, как долго он раскручивается и почему хорошо обеспеченный ресурсами конкурент, стартовавший сегодня, не смог бы воспроизвести его меньше чем за два года.

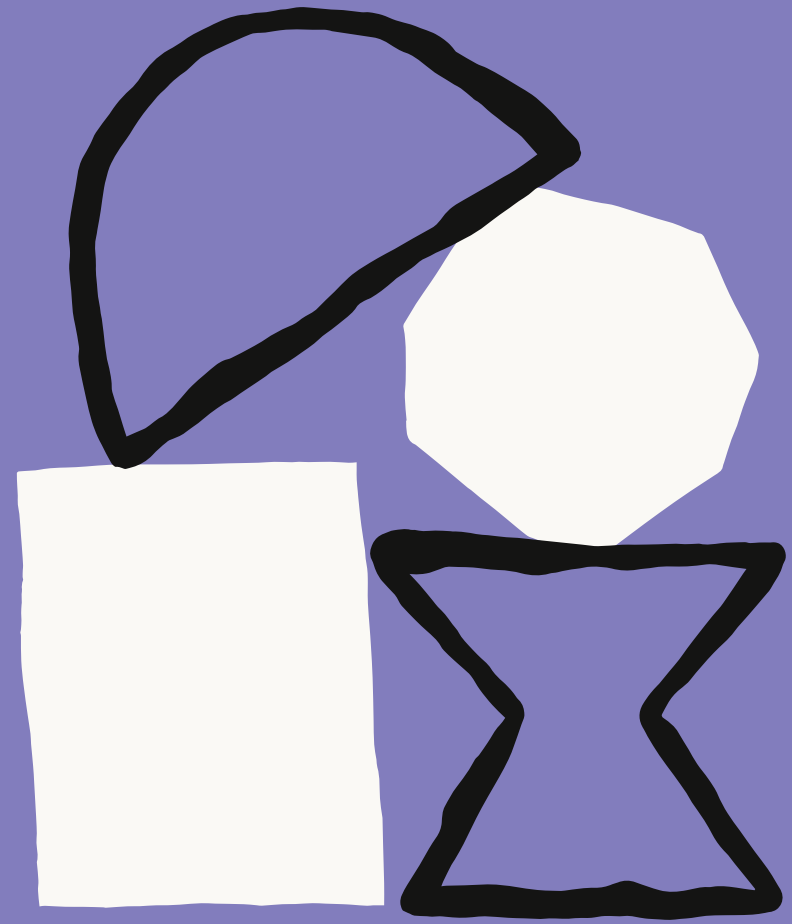
Создайте замыкание на рабочих процессах

Нарастающие сложным процентом сетевые эффекты данных делают ваш продукт труднее воспроизводимым, а замыкание пользователя на рабочих процессах делает ваш продукт труднее покидаемым. Чем дольше пользователи прогоняют ваш продукт внутри своих повседневных операций, тем глубже он встраивается в то, как они реально работают. Они построили поверх него автоматизации, обучили людей им пользоваться и подключили его к своим источникам данных и другим инструментам. Промпты, которые они разработали, рабочие процессы, которые они отточили, и результаты, которые они стандартизировали, — всё это сформировалось вокруг того, что делает ваш продукт и как он это делает. В этот момент переход на другое решение превращается из продуктового решения в полномасштабный операционный проект.

Первый шаг в создании замыкания на рабочих процессах — попросить Claude составить карту вашей текущей клиентской базы по глубине интеграции. Для каждого сегмента клиентов определите, какие рабочие процессы они построили поверх вашего продукта и от каких интеграций они зависят. Это показывает, где ваш продукт «прилипает», а где ему нужно уйти глубже.

Чем больше интеграций вы предлагаете, тем больше поверхности у клиента, чтобы выстраивать рабочие процессы, опирающиеся на ваш продукт. Claude Code помогает вам быстро поднимать нативные интеграции с конвейерами данных, инструментами управления проектами и другими системами, на которые полагаются ваши целевые пользователи. Claude Code также может построить API, вебхуки и SDK, которые позволяют клиентам не просто пользоваться вашим продуктом, но и строить поверх него, — глубочайшая форма замыкания.

- **Упражнение:** Попросите Claude помочь вам построить аудит интеграции рабочих процессов для ваших десяти ключевых клиентов. Для каждого задокументируйте автоматизации, которые они построили, интеграции, от которых они зависят, командные рабочие процессы, проходящие через ваш продукт, и вашу оценку стоимости их перехода на другое решение. Затем попросите Claude выявить паттерны по всей группе: какие типы интеграции создают глубочайшее замыкание для вашего конкретного продукта и что вы могли бы построить или включить, чтобы углубить интеграцию для клиентов, которые сейчас на поверхности.



Глава 7

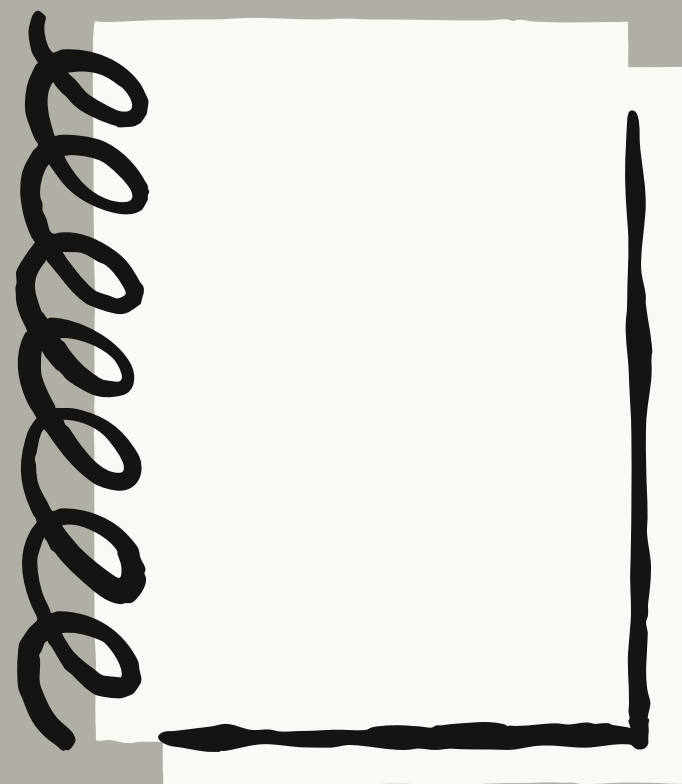
Та же работа, новые правила

Та же работа, новые правила

В эпоху ИИ работа основателя не изменилась: найти реальную проблему, построить то, что её решает, и масштабировать это в компанию, которая имеет значение. Изменился путь к этому. На всех четырёх стадиях — Идея, MVP, Запуск и Масштабирование — ИИ сжимает кварталы в недели.

Циклы валидации, которые раньше занимали месяцы, теперь занимают вечера. Рабочий прототип больше не требует сооснователя с нужным стеком; он требует ясной проблемы и нескольких сфокусированных сессий с кодинг-агентом. Готовность к запуску сжимается из предзапусковой суматохи в непрерывный рабочий поток. А на стадии масштабирования операционный вес, который раньше загонял первых нанятых в роль пожарных, всё больше можно передать ИИ, освобождая вашу команду тратить внимание на те суждения, которые становятся вашим защитным рвом.

Узкие места теперь — не то, что вы можете построить, а то, что вы выбираете строить.



Ресурсы

Ресурсы

Создание с Claude

- **«Building AI Agents for Startups»:** рассказывает, как стартапы используют агентов, чтобы по мере роста становиться менее зависимыми от основателя.
- **«Claude Code docs»:** проводит создателей от начальной установки до продвинутых агентных рабочих процессов. Совет: начните с обзора «How Claude Code works».
- **«Claude Code best practices»:** охватывает паттерны, которые сработали внутри Anthropic и в разных инженерных командах, — управление контекстом, права доступа, планирование и процессы верификации.
- **«Using CLAUDE.md files»:** проводит через то, как настроить Claude Code под вашу конкретную кодовую базу. Обязательное чтение для основателей на стадии MVP, настраивающих свою среду разработки.
- **«Claude Code power user tips»:** освещает паттерны рабочих процессов от самой команды Claude Code, включая параллельные сессии и циклы верификации.
- **«Get started with Claude Cework»:** рассказывает, как команды могут настроить Claude Cework и начать внедрять навыки, плагины и другие возможности, масштабирующие его влияние на ваш стартап.
- **«Tutorials»:** claude.com/resources/tutorials предлагает доступный для поиска список практических разборов конкретных задач.
- **Carta Healthcare** использует Claude для своей платформы клинической абстракции, обрабатывая 22 000 хирургических случаев в год и сокращая время абстракции данных на 66%.
- **Anything**, работающий на Claude и Agent SDK, помог 1,5 миллиона пользователей превратить идеи в работающие программные продукты без написания кода — включая нетехнического основателя, который построил и уже продаёт полноценную рекрутинговую платформу. ИИ-агент Anything берёт на себя всю сборку, чтобы соло-предприниматели могли сделать ставку на свою экспертизу в предметной области.
- **Cogent** — это прикладная ИИ-лаборатория, строящая агентов для автоматизации критически важных задач корпоративной безопасности. Стартап использует Claude как слой рассуждений для агентов, автоматизирующих расследование, приоритизацию и устранение на протяжении всего жизненного цикла уязвимостей.
- **Airtree** использует Claude Cework как центр своей операционной инфраструктуры, объединяя данные, которые раньше были разбросаны по десятку разных инструментов и команд. Теперь, когда один человек строит автоматизацию рабочего процесса с помощью навыков, каждый в организации может использовать её, чтобы делать все те дела из списка задач, до которых раньше не доходили руки.
- **Duvo** строит ИИ-агентов, которые ведут процессы закупок, цепочек поставок и категорийного управления через ERP, порталы поставщиков, таблицы, почту и даже телефонные звонки. Duvo построен целиком на Claude, используя Agent SDK для оркестрации рабочих процессов.

Истории основателей

- **«How three YC startups built their companies with Claude Code»:** разбор того, как HumanLayer (F24), Ambral (W25) и Vulcan Technologies (S25) использовали Claude, чтобы быстро вывести прототипы на рынок и масштабировать ИИ-платформы с помощью агентных процессов разработки.
- Основатели **GC AI** использовали экспертизу в предметной области, чтобы построить отзывчивую юридическую платформу на базе Claude — под то, как внутренние команды реально работают: специфичные для компании плейбуки, кросс-функциональные стейкхолдеры и переменные пороги толерантности к риску.
- **Zingage** — это платформа ИИ-агентов, построенная для круглосуточных автоматизированных операций агентств по уходу на дому. Стартап использует структурированные вызовы инструментов Claude для оркестрации между EMR и множеством каналов связи, а контекстные рассуждения Claude — чтобы строить агентов, способных давать тонкие, подстроенные под пациента результаты, а не подбирать наиболее частый шаблонный ответ.
- **Kindora** — это платформа на базе ИИ, построенная руководителем некоммерческой организации, который использовал Claude Sonnet, чтобы создать остро необходимый инструмент для умного подбора благотворительных организаций и доноров. Отфильтровал тысячи совпадений до

немногих, достойных внимания, MCP-коннектор Kindora позволяет некоммерческим организациям обращаться к её инструментам поиска прямо внутри Claude.

- **Wordsmith** была основана юристом, ставшим СТО, чтобы предоставлять надёжные юридические технологии на базе ИИ для внутренних юридических команд. Claude — это движок рассуждений для возможностей Wordsmith по проверке договоров, составлению соглашений и ревью документов, а инженерная команда стартапа использует Claude Code для построения и развития самой платформы.

Поддержка стартапов и возможности

- **Anthropic Startups Program:** для стартапов, работающих с венчурными партнёрами Anthropic, программа предоставляет бесплатные API-кредиты, высший уровень публично доступных лимитов запросов и приглашения на эксклюзивные мероприятия и воркшопы для основателей.
- **Сообщество Claude:** форумы и пространства сообщества для создателей.
- **Ресурсы для живого обучения:** конференции, вебинары, прямые трансляции и записи.



claude.ai