



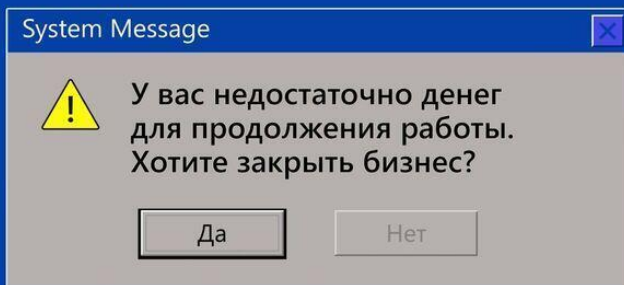
18+

Ваша команда совершила критическую ошибку
и должна быть уволена.

НЕ ВЗЛЕТИТ

РУКОВОДСТВО КАК СЖЕЧЬ

[МИЛЛИОН] НА МУР



STATUS: ACCESS_DENIED // ВОЗРАСТ: 18+

WARNING: СОДЕРЖИТ НЕЦЕНЗУРНУЮ БРАНЬ

AUTHOR: MAX DATA

Max Data

Не взлетит. Руководство как сжечь миллион на MVP

<https://litres.ru/74464863>

ISBN 9785007115537

Аннотация

Как убить IT-проект? Легко. Дайте расплывчатое ТЗ, всезнающего гения без коммуникации и менеджера, который «верит в команду». Эта книга — не про Успешный успех. Это чек-лист провалов, абсурдных решений, выгораний, бессмысленных переделок и вашего бюджета, сгоревшего в аду «гибкой разработки». Если вы инвестор, СЕО или всё ещё верите в лёгкие миллионы на разработке ПО — читайте, чтобы не вляпаться в разрушительные мечты, фатальные ошибки и ноль на балансе.

Книга содержит нецензурную брань.

Содержание

ВСТУПЛЕНИЕ. АНАТОМИЯ ТИПИЧНОГО ПРОВАЛА	7
Кому особенно стоит её прочитать?	14
Кто не поймет книгу — и пусть не пытается	17
Почему эта книга?	19
ЭТО ВСЁ ОЧЕНЬ ДОРОГО	22
DREAM TEAM	32
Кровавые процессы	40
Элитный заповедник, или Зачем проекту пытливые умы	45
Конец ознакомительного фрагмента.	47

Не взлетит Руководство как сжечь миллион на MVP

Max Data

© Max Data, 2026

ISBN 978-5-0071-1553-7

Создано в интеллектуальной издательской системе Ridero

Небольшое предупреждение для здравомыслящих читателей

В тексте этой книги вам неизбежно встретятся слова вроде *Блокчейн*, *ИИ*, *Web3*, *Метаверс* или *AR/VR*. Не пугайтесь. Это не ошибка редактора и не попытка автора казаться ультрасовременным. Это суровая реальность ИТ-индустрии, в которой абсолютно любой проект рискует в одночасье стать лабораторией для обкатки «передовых технологических решений». Разумеется, исключительно за счет бюджета самого бизнеса.

Знаете вы, что означают эти аббревиатуры, или нет — не имеет ровным счетом никакого значения. Главное, что вам нужно усвоить: как только на общем созвоне звучит призыв

срочно внедрить очередную инновацию, в девяти из десяти случаев речь идет не об улучшении продукта. Речь идет об удовлетворении чьего-то личного технического любопытства за чужие деньги. И далеко не факт, что авторы этой гениальной идеи потом согласятся взять на себя ответственность за неизбежные последствия.

То же самое касается и прочего профессионального сленга: *нетворкинг*, *стартап-питчи*, *воркшопы*, *гроусхакинг* и остальной словесной лапши из модных англицизмов. Если этот птичий язык режет вам слух — не переживайте, это здоровая реакция организма. Наличие подобной лексики в книге — не самоцель, а наглядная иллюстрация того, как в индустрии принято создавать ауру глубокой компетентности. Большинство этих терминов придуманы не для ясности коммуникации, а исключительно для того, чтобы с важным видом транслировать иллюзию глубокого смысла там, где его никогда не было.

Вы непременно заметите на этих страницах опечатки, шероховатости стиля, где-то лишнюю запятую, а где-то — легкое нарушение логики изложения. Относитесь к этому спокойно. В этой книге, как и в абсолютно любом коммерческом ИТ-продукте, гарантированно присутствуют баги, архитектурные костыли и контролируемая доза хаоса. И дело вовсе не в лени автора — просто безупречный идеал является мифом. Особенно в нашей индустрии.

В реальной жизни ошибаются абсолютно все: опытные ар-

хитекторы, ведущие разработчики, дотошные тестирующие, генеральные директора и многоопытные инвесторы. Поэтому, если по ходу чтения вам что-то покажется странным или спорным, не спешите раздражаться. Возможно, перед вами обычный баг, а возможно — тщательно спроектированная фича. Подобные книги создаются не в порыве романтического вдохновения на вилле у моря, а в редких перерывах между работой, либо чаще всего ночами.

«СПАСИБО ВСЕМ, КТО ПРОФИНАНСИРОВАЛ МОЙ ОПЫТ»

ВСТУПЛЕНИЕ. АНАТОМИЯ ТИПИЧНОГО ПРОВАЛА

Всё всегда начинается одинаково красиво: собирается группа инициативных людей и решает запилить великий ИТ-продукт. Кто-то нашел стартовый капитал, кто-то привел «знакомых пацанов-разработчиков», кто-то клятвенно пообещал космический результат. Начинается бурное движение: бесконечные созвоны в Зуме, плодятся чаты в Телеграме, забиваются мусором Google Docs. Все бегут марафон за MVP. На коленке верстается лендинг, прикручивается сгенерированный нейронкой логотип, дизайнеры что-то усердно рисуют, программисты что-то бешено кодят. Все безумно заняты. Иллюзия бурной деятельности стопроцентная.

Но проходит месяц, и картинка начинает предательски плыть. Выясняется, что внятных требований к продукту нет, четкие роли в команде отсутствуют, дедлайны взяты исключительно с потолка, а приоритеты меняются трижды за день прямо во время обеда. Деньги инвестора улетают в трубу с такой скоростью, будто главная цель проекта — сжечь бюджет до конца квартала. Инженеры работают вхолостую, выдавая тонны сырых прототипов, которые никто и никогда не валидировал на реальном рынке. Коммуникация превращается в хаотичный базар, ключевые решения принимаются по

настроению руководства, а одни и те же грабли бьют команду в лоб с завидной регулярностью.

Еще через месяц наступает финальная стадия отрицания реальности. Половина написанных функций нахер никому не нужна, а вторая половина попросту не работает. Бюджет превратился в пепел. У заказчика — паническая атака, у разработчиков — глубокая клиническая апатия. Самые шустрые начинают тихо ливать из проекта. Те, кто остался, продолжают просто сидеть на зарплатной игле, полностью потеряв остатки веры. Исход предсказуем: вместо масштабируемого продукта на выходе получается невнятное цифровое нечто, а вместо команды — случайный набор демотивированных людей, которые просто тупо доживают до очередного дедлайна.

Это и есть реальный запуск большинства IT-проектов. Всё, что теоретически могло пойти не так, пошло именно так. Потому что с самого начала никто не подумал о простом: прежде чем начинать, нужно договориться, зачем, для кого и как. Всё остальное — лишь закономерные следствия. Следствия, которые неизбежно порождают элитную «команду ликвидаторов» вашего бизнеса.

Знакомьтесь, спонсоры горящего продакшена:

Project Manager (PM). Пытается управлять сложной технической разработкой с помощью экселевских табличек, кофейной гущи и собственной интуиции. Руководит процессами исключительно по наитию, путая управление проектами

с микроменеджментом и бесконечной рассылкой писем.

QA-инженер. Подключился к процессу, когда продукт уже благополучно уехал в продакшн. Зачем спешить, ведь «всё же и так работает» (или работало?), да и вообще тесты — это долго.

Техлид. Человек, который в реальной жизни никогда не доводит до конца ни один личный проект, но зато смело берётся проектировать высоконагруженную и масштабируемую архитектуру для вашего бизнеса.

СЕО. Для него MVP — это просто модное слово из презентаций, не имеющее никакого отношения к реальности. Продуктовое видение балансирует на космическом уровне «что-то между CRM, GPS и немного NLP».

Феномен MVP: Жизнеспособность против амбиций

Давайте начистоту про MVP. В идеальном мире, который показывают на конференциях, MVP — это *Minimum Viable Product*, то есть минимально жизнеспособный продукт. Подчеркнем: именно *жизнеспособный*, а не идеальный. Его цель — не покорить рынок с первого релиза, не впечатлить инвестора параллакс-эффектом и не победить в номинации «Лучший дизайн». Его единственная задача — проверить гипотезу, быстро и с минимальными затратами. Узнать, есть ли у идеи пульс, или она — ещё один стартап-призрак на бескрайнем ИТ-кладбище.

Правильный MVP монументален в своей простоте:

Стабильный: он не крашится, не съедает оперативную

память и не убивает сессию пользователя при первом клике.

Минимальный: в нем ровно столько функций, чтобы пользователь понял суть и смог дать осознанный фидбэк.

Жизнеспособный: если этим решением невозможно пользоваться для закрытия базовой боли — это не MVP, это просто скриншот экрана из Figma.

Он не обязан поражать воображение красотой, но он обязан работать. Он не обязан быть полным, но обязан решать одну понятную задачу для одной понятной аудитории. Остальное — потом. Если оно вообще кому-нибудь понадобится.

В суровой реальности MVP мгновенно превращается в огромную, раздутую от управленческих амбиций тушу, над которой команда парализованно корпееет по 8 месяцев. А всё потому, что на общих митингах начинают звучать гениальные аргументы:

«Ну как без авторизации через Telegram и анимированных стикеров? Без этого я бы точно не пользовался!»

«Ну нельзя же показывать людям сырой продукт, у нас страдает бренд-вижн!»

«Ну давайте ещё сделаем дашборд с аналитикой, ИИ-рекомендации и автозаполнение форм на случай, если кто-то случайно зайдёт!»

И вот MVP из инструмента быстрой проверки гипотез превращается в многомесячный долгострой. Разработчики бесконечно полируют никому не нужный функционал, ди-

зайнеры рисуют девятый вариант интерфейса, а руководство смотрит на *burn rate* (темп сжигания денег) и медитативно шепчет: *«Ещё чуть-чуть... вот сейчас допилим профиль пользователя и точно выстрелит»*.

Но продукт не выстреливает. Потому что всё это время вы строили идеальный самолет, даже не зная — нужен ли людям вообще перелёт. Может, ваша целевая аудитория хотела лодку. Или им было вполне достаточно обычного велосипеда.

Есть известное и очень циничное высказывание: *Если вы вышли на рынок с продуктом, который вас полностью устраивает — вы опоздали*.

И знаете почему? Потому что перегретый рынок не ждет, пока вы причесываете кнопки, настраиваете отступы и проверяете пиксели. Пока вы тратите месяцы на полировку очередного идеала, ваш более приземленный конкурент уже выкатил кривой, косой прототип, проверил на нем ключевую гипотезу, собрал первую обратную связь, сделал работу над ошибками и прямо сейчас уводит ваших потенциальных клиентов.

Быстро, просто, стабильно — вот три слона настоящего MVP. Всё остальное — из другой индустрии, например, кинематограф. Или от перфекционизма, синдрома отличника и панического страха, что кто-то в комментариях скажет «фу».

И знаете что? Скажут в любом случае. Даже если продукт

будет идеальным. Только вот проблема: вы можете потратить год жизни и миллион долларов, чтобы услышать то, что могли бы бесплатно узнать за первую неделю.

Зачем написана эта книга?

Потому что автору уже патологически больно смотреть, как умные, взрослые люди раз за разом наступают на одни и те же ржавые мины, на которые просто наклеили новые глянцевые стикеры. Я своими глазами видел, как бесследно сгорают огромные бюджеты, как прагматичные инвесторы от безысходности превращаются в философов, а талантливые разработчики — в выжженных, апатичных биороботов, у которых на лице застыла единственная эмоция: *«Мы это уже проходили. Пожалуйста, не надо»*.

Эта книга — не очередная скучная инструкция по достижению успеха. Это жесткая антиинструкция. Практическое руководство по тому, как делать точно не надо. Мы подробно препарируем самые популярные системные ошибки: от слепого клонирования хайповых западных трендов до наивной веры в то, что аутсорс-разработка — это магия, а не лотерея с отрицательным математическим ожиданием. Здесь не будет политкорректных сглаживаний углов. Будет цинично, местами грубо, но всегда честно. Мы поговорим о том, кто такие токсичные инвесторы, зачем в проекте нужен третий кофаундер с «особенным взглядом на вещи», и почему фраза *«надо просто нормально написать»* звучит как смертный приговор для продукта.

Если вы СТО, стартапер, менеджер или просто человек, которому однажды сказали: *«Ты же айтишник, сделай нам приложение»*, эта книга — про вас. Добро пожаловать в мир, где MVP — это не продукт, а ритуал красивого самоубийства бюджета.

Да, временами может показаться, что всё происходящее в индустрии — абсолютный абсурд. Будто проекты строятся не благодаря усилиям, а вопреки здравому смыслу. И что каждое успешное приложение — случайная мутация в хаосе продуктовой эволюции. Но это не так.

В ИТ работают толковые, талантливые, часто гениальные люди. Именно благодаря им мы пользуемся сервисами, которые стали частью повседневной жизни. Просто иногда процессы заносит на поворотах. Управленец внезапно решает, что без AR в приложении для заметок «не будет вау-эффект». Разработчик пишет код, который через месяц не сможет объяснить даже самому себе под пытками. Заказчик третий раз подряд полностью меняет ТЗ, потому что *«ну, я же плачу деньги — пусть всё будет идеально»*.

Всё это не трагедия — это реальность разработки. Не тотальный бардак, а скорее повторяющийся цикл ошибок, который можно прекратить, если вовремя остановиться и спросить: *«А зачем мы это делаем?»*. Проблема не в людях — проблема в паттернах. И именно с ними мы и будем разбираться дальше.

Кому особенно стоит её прочитать?

Инвесторам, у которых горят деньги, но не светятся глаза

Вы вложили средства в «цифровой Uber для доставки воды с искусственным интеллектом», а в итоге получили одностраничный лендинг и двоих кофаундеров, живущих на Ба-ли и постящих мудрые цитаты в соцсетях? Эта книга поможет вам узнать, куда именно испарился ваш миллион и кто из команды отвечал за это испарение. Спойлер: никто.

Вы поймёте, как выглядит настоящий продукт, а не фантастическая декорация для питча. Научитесь отличать разработку от имитации, а «стадию pre-pre-MVP» — от полной некомпетентности.

Руководителям и фаундерам, особенно тем, кто «в бизнесе, а не в коде»

Если вы мечтаете «запустить стартап» и «нанять сильную команду», но не знаете разницы между фронтендом и фреймворком — читайте дважды. Вы узнаете, почему нельзя ставить сроки без декомпозиции, почему друг-дизайнер не должен автоматически становиться вашим СРО, и почему не бывает команды за копейки, которая сделает вам работающую криптобиржу. Эта книга научит вас главному: честно называть вещи своими именами, даже если это больно. Особенно если больно.

Техлидам и СТО, которым приходится тушить пожары вместо того, чтобы строить архитектуру

Вы прекрасно знаете, как выглядит «готово» на словах менеджера, и «готово» по факту в репозитории. Вас регулярно заставляют писать план релиза для фичи, которую придумал маркетолог после бокала просекко? Тогда эта книга — как холодный душ после грязного спринта. Здесь вы найдёте аргументы, которыми можно культурно объяснить бизнесу, почему нельзя сделать всё и сразу. И как выглядят настоящие технические решения, а не фэнтези-презентации в PowerPoint.

Менеджерам проектов, которые зажаты между продуктом, багами и бессонницей

Вы ежедневно балансируете между бизнесом, который хочет всё ещё вчера, командой, которая не понимает «зачем это вообще писать», и заказчиком, у которого «срочно» — это базовый стиль жизни? Эта книга для вас. Мы обсудим, как не превратиться в курьера по перекладыванию тасок, как выжить между двух огней и не сойти с ума от Jira, где каждая задача помечена как «критический приоритет».

Всем, кто хоть раз запускал ИТ-проект и говорил себе: «Больше никогда»

Потому что вы всё это уже видели. Вы прекрасно знаете, как реальность отличается от красивых TED-токов. Вы лично слышали эти вечные мантры:

«Нам нужно просто найти нормальных разработчиков»

«А можно MVP до конца недели? Там же всего две кнопки»

«Давайте просто возьмем дешевый аутсорс — они обещают сделать быстро»

Вы прожили это и знаете, какой тяжелый путь рождения проделывает та самая кнопка на экране смартфона, которую конечный пользователь, возможно, никогда и не заметит. Эта книга написана для того, чтобы вы хотя бы посмеялись. Или заплакали. Но в любом случае — не чувствовали себя одинокими в этом цифровом хаосе.

Кто не поймет книгу — и пусть не пытается

Если вы искренне считаете, что «айтишники просто ленивые и всё делают долго» — немедленно закройте этот текст. Вам лучше читать мотивационные цитаты успешных людей, смотреть рилсы про личностный рост и заниматься нетворкингом. Тут не будет волшебства, только системный подход, прагматичный цинизм и горький опыт.

Если вы свято верите, что можно нанять студента за еду и сделать аналог Тинькофф-банка за 3 месяца — оставьте надежду. Вы не фаундер. Вы — разносчик пустых обещаний и коллекционер дорогостоящих факапов. И эта книга — не учебник, а предупреждение для тех несчастных, кто ещё не пошёл за вами.

Если вы менеджер, у которого на любые проблемы один ответ: *«Всё будет нормально, не переживайте»* — пожалуйте, начните переживать. Ваши проекты — это братское кладбище идей с красивыми названиями. Эта книга — ваша персональная табличка с надписью: *«Здесь покоится MVP, убитый завышенными ожиданиями»*.

Если вы из тех, кто привык повторять: *«Тут нужен просто хороший специалист, и всё заработает»* — не читайте дальше. Эта фраза звучит каждый раз перед тем, как компания нанимает пятого тимлида подряд, потому что «что-то с

прошлыми было не так». Может, дело всё-таки не в них?

Почему эта книга?

Когда я только входил в ИТ, мне казалось — всё давно расписано и структурировано. Методологии, умные книги, лучшие мировые практики. Следуй Agile, читай Scrum-гайд, внедри DevOps — и понеслась. Успех гарантирован, как в рецепте из кулинарной книги: просто сделай всё строго по пунктам.

Но реальность, как обычно, оказалась гораздо веселей.

Все эти идеальные советы, что пишутся в глянцевах учебниках и на модных курсах, работают при одном маленьком условии — если вы живете в абсолютном вакууме фантазийного стартапа из рекламного ролика. Например, в таком:

Очередь из гениев. У вас под дверью стоит очередь из сильнейших разработчиков, мечтающих работать за идею, туманные токены и простое человеческое спасибо. Они не требуют рыночной зарплаты, не выгорают, и все, как один, наизусть знают Clean Architecture и Rust.

Бесконечный бюджет. Финансовая подушка не имеет дна. Офис с панорамным видом, топовое железо у каждого сотрудника, кофе из зерен, собранных вручную на склонах Гватемалы. И три вида альтернативного молока — потому что разнообразие, как известно, залог стабильности продукта.

Идеальная команда. Никто не уходит в отпуск за день

до важного релиза, все обожают друг друга на ретроспективах, и ни у кого в тайных планах нет идеи сбежать в конкурентный стартап, прихватив с собой полный бэкап базы данных.

Отсутствие конфликтов. Все мило улыбаются, поддерживают любые инициативы, уважают чужие дедлайны и принципиально не трогают продакшн по ночам. Тимбилдинги проходят цивилизованно, без драк, пьяных откровений и последующих увольнений.

Зарплатное табу. Уровень компенсаций — тема священная. Никто не знает, кто сколько получает, и никого это не волнует. Все находятся на одном высоком духовном вайбе — ведь мы команда, а не банальный рынок труда.

А потом приходит реальная жизнь. И внезапно оказывается:

Scrum с грохотом ломается о самоуправляемую токсичную команду.

Продуктовый «визионер» хочет всё, сразу и желательно бесплатно.

Технический долг накапливается быстрее, чем вы успеваете гуглить, что это слово вообще означает.

И весь ваш амбициозный проект начинает напоминать сериал, где первый сезон был многообещающим, а дальше — полный провал, хаотичная смена актеров, падение рейтингов и резкий, позорный финал.

Эта книга именно про то, что происходит, когда красивые

книжные правила сталкиваются с реальными людьми, ограниченными бюджетами, раздутым эго, горящими дедлайнами и неработающим Git. Когда всё идёт не по плану. А это происходит почти всегда.

Если вы хотите понять, почему всё сыпется, как не сжечь свой проект на старте и не попасть в ИТ-мемы на очередной конференции — читайте. Если хотите сделать работающий продукт, а не получить «бесценный опыт» за сотни тысяч инвесторских долларов — читайте. Если просто хотите поржать над чужими фейлами и узнать, почему вы не один такой в этой лодке — тоже читайте.

А если вы хотите разрушить свой проект максимально красиво, эффектно и с огоньком — следуйте этому тексту как пошаговому официальному мануалу. Здесь бережно собраны все чужие ошибки, чтобы вам не пришлось изобретать их заново.

ЭТО ВСЁ ОЧЕНЬ ДОРОГО

Если вы думаете, что IT-проект — это путь к миллионам, остановитесь. На старте это не бизнес. Это **дыра**, в которую вы весело и с энтузиазмом кидаете деньги. И чем громче презентация, тем глубже дыра.

С первого же дня — **минус**. Купили первый ноутбук (который, кстати, стоит как годовая зарплата стажера)? Минус. Наняли разработчика (который, скорее всего, уйдёт через полгода)? Минус. Арендовали офис, заказали кофе и подписались на Jira? Поздравляем, вы основали предприятие по сжиганию денег.

Пока вы ещё на бумаге рисуете «юнит-экономику» и «точку безубыточности», **реальность уже забрала ваш MacBook, часть команды и два месяца аренды.**

Запуск IT проекта — как запуск ракеты. Только вместо керосина — деньги инвестора, вместо инженеров NASA — вы, Google и дедлайн (который наступил уже позавчера). Ошибся с выбором библиотеки? Плюс две недели. Ошибся с наймом? Плюс три месяца. Всё это оплачивается. Зарплатой, арендами, нервами. Ошибки здесь — **не баги, а статьи расходов.**

Офис: зло, но необходимое

Давайте сразу разберемся с этой вашей священной коровой. Да-да, «мы все сейчас на удалёнке», «у нас прогрессив-

ный гибрид», «асинхронная кросс-функциональная команда» и прочий приторный смузи-шейкер из Линкедина. Но пока вы сладостно мечтаете о безупречном виртуальном метапространстве, ваша реальная команда эпически косячит в Слаке, отвечает на судьбоносные митинги с заспанной рожей прямо из-под одеяла и умудряется синхронизировать простейшую задачу по четыре дня, потому что они, видите ли, раскиданы по разным часовым поясам и ментальным орбитам.

На старте любого стартапа физический офис — это не пережиток прошлого и не корпоративная блажь, а единственный доступный вам ускоритель процессов. Вам жизненно необходимо коммуницировать. Лично. Лицом к лицу. Ругаться, спорить до упора, на ходу генерировать гениальный бред и тут же разносить его в хлам. А под конец недели — вместе бухать в ближайшем пабе. Уважаемый HR-департамент, выдохните и не дергайтесь, это просто метафора. *Или нет.* Решайте сами.

Но тут всплывает суровая бытовая правда: чтобы заставить зажавшегося айтишника выползти из зоны комфорта и добровольно вернуться в опенспейс, ваш офис по уровню уюта должен как минимум превосходить его съемное жилище. Начнем с того, что поясница сеньор-разработчика — это неприкосновенно, так что раскошеливаемся на эргономичные кресла по хорошей цене. Железо тоже должно быть топовым: забудьте про старые Макбуки на процессорах Intel,

ведь любой уважающий себя джун с ходу заявит, что на этом доисторическом компе у него даже не открывается VS Code.

Сюда же добавляем расходы на собственное серверное железо, потому что на старте вам обязательно покажется, что раздутое облако Амазона — это «необоснованно дорого». Ха-ха. Подождите радоваться экономии, вы просто еще не видели, сколько в этом году просят за выкуп красивого доменного имени в зоне. com.

Разумеется, вам придется организовать бесперебойные поставки кофе, чая и печенек. И упаси вас бог забыть про три альтернативных вида молока: если на кухне внезапно закончится кокосовое или миндальное, ваш ведущий UI/UX-дизайнер демонстративно соберет вещи и уйдет к конкурентам прямо посреди спринта.

Вам критически необходима игровая зона с PlayStation, причем исключительно для того, чтобы вы, как фаундер, могли по логам безошибочно вычислять, кто в этой компании вообще не работает (возможно, это будут все). Наземная парковка тоже обязательна, иначе половина штата просто откажется доезжать до работы. На сдачу наймите толкового офис-менеджера, потому что без этой няньки ваши суровые инженеры тупо умрут от голода посреди нераспакованных коробок с доставкой еды. Ну и вишенка на торте — полноценный бар прямо в переговорке, без которого всю эту ИТ-историю ни один нормальный человек психически просто не выдержит.

Ах да, чуть не забыл про налоги. Государство будет незримо стоять за вашим плечом каждую секунду, регулярно и молча забирая свою долю. Так что привыкайте. Считайте, что налоговая инспекция — это ваш самый главный, самый молчаливый и самый неотвратимый сооснователь бизнеса.

Давайте добавим еще немного суровой, неприкрытой реальности. Ниже — классический шорт-лист тех вещей, о которых вы в своей фаундерской эйфории даже не задумывались, но они все равно придут за вашими деньгами.

Облака, которые вам гарантированно не по карману.

«AWS — это дешево и масштабируемо», вы определенно где-то прочитали в интернете. Дааа, охотно верю. Ровно до тех пор, пока весь ваш продукт представляет собой статичный лендинг на три картинки, сиротливо лежащий в S3-бакете. Но стоит вам дрожащей рукой нажать кнопку «Деплой в продакшен», как в начале следующего месяца вам прилетает инвойс на пять тысяч долларов. За что? Почему? Никто в компании понятия не имеет. Мигрировать на другое облако? Поздняяк, это еще дороже. Оптимизировать архитектуру? Некому, ваши вайб-кодеры умеют только копипастить из Чата ГПТ.

В итоге вы просто покорно платите, потому что отдавать по пять штук баксов ежемесячно тупо проще, чем попытаться понять, какая именно скрытая метрика минусует ваш бюджет.

Лицензионный ад, или плати за воздух.

Вы наивно полагали, что софт для разработки покупается один раз и навсегда? Добро пожаловать в прекрасный мир подписочной модели, где вы не владеете вообще ничем. Приготовьтесь каждый год отстегивать по пятьсот долларов за профессиональные IDE для каждого инженера, кормить Фигму еще на пару сотен за каждого причастного к дизайну и отдавать почти тысячу долларов в год за корпоративный Слак исключительно ради того, чтобы ваши бездельники могли лепить кастомные анимированные эмодзи в рабочих чатах. Сюда же поверх ложится годовая лицензия на тяжелый Фотошоп, просто потому что вашему дизайнеру «так привычнее и надо для вдохновения». Самое смешное, что половиной из этого оплаченного софта ваша команда не воспользуется ни разу, но счета будут приходить строго по расписанию.

DevOps, или плата за абсолютное ничегоделание (Dolce far niente).

Менеджмент упорно не понимает, за что этому человеку отваливают деньги, ведь со стороны он выглядит как самый ленивый персонаж в офисе, который просто пьет кофе и меланхолично смотрит в монитор (если он ходит в офис вообще). Но математика здесь парадоксальная: если ваш девопс круглосуточно суетится, с красными глазами чинит сервера, вручную перезапускает контейнеры перенастраивает CI или Графану (Grafana) — поздравляю, у вас на проекте катастро-

фа. Нормальный DevOps-инженер инвестирует интеллект в то, чтобы один раз настроить автономную экосистему, которая бесперебойно живет в сети, сама масштабируется при наплыве пользователей, автоматически реанимирует себя в случае сбоев и работает вообще без его участия. Вы платите ему оклад именно за то, чтобы он сидел сложа руки. И если в вашем проекте от него наступила идеальная, звенящая тишина — это значит, что ваш невидимый страж на бэкграунде гениально выстроил автоматизацию. Но стоит вам сэкономить на этом «бездельнике», и первый же косяк упавшей инфраструктуры выставит вашему бизнесу brutalный счет. Нет продукта в сети = нет продукта в принципе.

Юристы как узаконенное вымогательство.

Правильное оформление юридического лица, NDA, кабальные договоры, пользовательские соглашения, политики обработки персональных данных, патенты и превентивная защита от патентных троллей — всё это стоит баснословных денег. Нанятый вами адвокат будет с умным видом брать по триста долларов за час своей неспешной работы, чтобы в конце недели выдать вам гениальное заключение: «Нет, ребята, называть ваше приложение iCoolApp нельзя, корпорация Apple засудит вас за две секунды». Ирония в том, что Apple про вас даже не знает, но триста долларов в час платите именно вы, а не Тим Кук.

Маркетинг, или искусство тратить.

На старте вам внушат, что без агрессивного продвиже-

ния, пиара и поисковой оптимизации ваш продукт — просто невидимый цифровой труп. Вы соглашаетесь и вступаете в игру. Что вы получаете на выходе? Помпезный корпоративный сайт за пять тысяч долларов, который не генерирует ни единой продажи. Поисковое SEO-продвижение за две штуки баксов в месяц, которое выведет вас на первую страницу Гугла примерно через год, когда компания уже трижды обанкротится. Трендовую видеорекламу в Тиктоке, которую лениво пролистали боты и которую не увидел ни один реальный живой человек.

И, конечно же, наемный топ-маркетолог, который за пару недель спустит ваш рекламный бюджет на десять тысяч долларов, после чего покажет фокус со своим исчезновением, бросив на прощание: «Ну а что вы хотели? Зато мы прокачали узнаваемость бренда, а прямые продажи вообще не входили в мои KPI»

Обучение персонала, или «вы и так дураки, но давайте доплатим».

Курсы повышения квалификации, модные тренинги личной эффективности и сертифицированные Agile-интенсивы — это лучший способ добровольно слить деньги в никуда.

Что вы получаете на выходе за свои кровные пять тысяч долларов? Красивую глянцевую бумажку в рамочке. А в придачу — проектного менеджера, который теперь на полном серьезе считает себя Scrum-коучем, и сеньор-разработчика, который после двухдневного вебинара в ультимативной фор-

ме требует немедленно переписать весь ваш работающий монолит на Go, иначе он потеряет ментальную связь с профессией.

Тестирование, или почему баг — это ваш новый финансовый долг.

Повторяйте за мной: QA — это не опция для богатых стартапов, это ваша единственная подушка безопасности перед лобовым ударом об стену. Без дотошного тестировщика весь ваш продукт гарантированно развалится на куски в день релиза. Математика здесь простая и жестокая: каждый вовремя найденный на этапе разработки баг экономит вам кучу денег. Но каждый баг, который просочился на прод и был обнаружен реальным пользователем, — это не просто ошибка. Это выставленный вам счет на сумму, которая способна загнать вашу компанию в кассовый разрыв.

Служба поддержки: за ваши же деньги вас будут искренне ненавидеть.

Как только продукт увидит свет, вам придется нанять людей, которые будут круглосуточно принимать ушаты помоев. Разгневанные пользователи начнут писать, звонить и яростно спамить во все каналы. «У меня ничего не работает!», «Где, блядь, эта кнопка?!», «Верните мои деньги, твари!». Вы будете ежемесячно платить зарплату саппорту только ради того, чтобы слушать, как клиенты хамят вашей компании в два часа ночи, обрывают телефоны по пять раз в день и шлют проклятия вперемишку с юридическими угрозами.

Привыкайте, для ИТ-бизнеса это абсолютно штатный режим функционирования.

Бухгалтерия как узаконенный и легальный грабёж. Приготовьтесь отдавать минимум штуку баксов в месяц толковому бухгалтеру просто за то, чтобы вас не посадили. Налогами в этой индустрии облагается вообще всё, что шевелится, а за то, что не шевелится — придется заплатить пошлину. Вы будете регулярно получать абсурдные штрафы от налоговой просто потому, что ваш ленивый менеджер в каком-то отчете «поставил галочку не в том поле». А еще вам придется часами безрезультатно доказывать собственному главбуху, почему ящик элитного виски для пятничного тимбилдинга — это «расходы на стратегическое развитие человеческого капитала», а не ваша личная прихоть.

Канцелярия, бытовуха и прочая офисная номенклатура. Разноцветные стикеры для Agile-досок — это скрытый наркотик для вашей команды. Без них современные айтишники физически не способны сгенерировать ни одной мысли. Цветные маркеры, маркерные доски, бумага для принтера, салфетки, туалетная бумага, питьевая вода и печенюшки.

На эти «бытовые мелочи» у вас будет стабильно улетать по триста-четыре доллара в месяц. Попробуйте ради эксперимента сэкономить и купить чай попроще — в офисе сразу начнутся стоны о том, что в таких скотских условиях «код не пишется и вообще ничего не работает».

Непредвиденные расходы — единственная постоянная статья вашего бюджета.

Запомните: Что-то обязательно сломается. Кто-то обязательно исчезнет. Кто-то обязательно придумает «ещё одну фишку за вчера».

И это не сбой системы. Это и есть глубинная суть информационных технологий. Форс-мажор в этой индустрии — не досадное исключение из правил, а ваш ежедневный, базовый режим работы.

Добро пожаловать в суровую реальность коммерческой разработки софта.

Именно поэтому любой успешный, прибыльный запуск ИТ-продукта на рынке — это не чудо, не гениальность менеджмента и не божественное провидение. Это всегда чудовищно дорогая случайность, методичная и дисциплинированная работа над преодолением полосы препятствий.

DREAM TEAM

Запомните: комплектование галеры всегда начинается с одного-единственного человека — с **Техлида**.

Без него у штурвала вашей разработки будет находиться либо случайный пассажир с купленным дипломом, либо звенящая пустота.

Правильный Техлид — это не тот персонаж, который на ежедневных утренних стендапах устраивает TED Talk. Это не болтун с горящими глазами, умеющий виртуозно вешать инвесторам лапшу на уши про блокчейн-революцию и квантовое будущее. Техлид — это циничный, хладнокровный хирург. Он смотрит на ваш кривой, скопипащенный из чат-бота код и, не моргнув глазом, достает скальпель. Он не цокает сочувственно языком — он молча чинит. Он принципиально не ведется на сиюминутную технологическую моду, он просто вовремя останавливает внутрикомандный идиотизм.

Если вы с гордостью несете ему в переговорку очередную гениальную мысль в духе: *«Слушай, а давай срочно бахнем в проект смарт-контракты, генеративный ИИ и что-нибудь из Web3!»* — правильный Техлид должен посмотреть на вас с такой глубокой, экзистенциальной скорбью, чтобы вам самому стало хоть немного стыдно за свое техническое слабоеумие.

Он на молекулярном уровне знает великую истину: иде-

альный код — это тот код, которого пока вообще не существует в природе. Потому что прежде чем писать терабайты символов, нужно сесть и четко понять, нахрена они вообще нужны твоему бизнесу. Его главный супернавык — это ювелирный баланс между *«так правильно и красиво по канонам архитектуры»* и *«так надо сделать прямо сейчас на костылях, потому что дедлайн был еще вчера. Потом исправим»*.

Техлид — это твой единственный живой буфер между работающим продуктом и неминуемой гибелью проекта от *feature creep* (бесконтрольного раздувания фич), фатальных багов, синдрома непризнанного гения у джунов и продукто-вика-гуманитария, который на ночь начитался Хабра и теперь с пеной у рта требует развернуть отдельный микросервис под каждый всплывающий попап на сайте.

Настоящих, трушных Техлидов на рынке критически мало. Практически все, кого вы встретите на собеседованиях это китайские подделки с накрученным опытом. Перепутаете, поведетесь на красивое резюме и получите обычную говорящую голову в оверсайз-футболке с надписью *«Senior»*. Этот персонаж будет с академическим, умным видом методично разрушать вашу кодовую базу, сливать сроки и называть этот откровенный саботаж «правом на инженерную свободу».

Но если вам крупно повезёт и вы умудрились нанять того кто вам действительно нужен, у вас появляется уникальный исторический шанс сжечь свой инвесторский бюджет не

впустую, а хотя бы с толком.

Запомните раз и навсегда: Техлид — это не статусная декорация для зум-митингов и не просто «самый старший разработчик по выслуге лет». Это единственный вменяемый человек в вашей конторе, который молча берет и делает правильно, пока вся остальная команда бесконечно спорит в чатах, меряется мемами в курилке и усердно пьет третий за утро латте на кокосовом молоке. (Или, судя по качеству их кода, там в чашках уже давно плещется чистый коньяк?).

Кровавый кастинг в дурдоме: Как нанять тех, кому на тебя насрать

Любая, даже самая гениальная, прорывная бизнес-идея остается просто сотрясением воздуха и красивой болтовней для инвесторов до тех пор, пока не появится команда, способная превратить этот бред в стабильно работающий софт. И ровно в этот момент на тлеющих руинах твоих фаундерских иллюзий начинается настоящее, первосортное трэш-реалити-шоу: с кастингами, взаимными проклятиями, нервными срывами менеджмента и сомнительными персонажами, которых тебе по дружбе притащил «один очень надежный знакомый».

Вообще, переход в любой новый проект — это всегда локальный, внутренний кризис. Вокруг тебя материализуются новые лица со своими странностями, на кофе-поинтах звучат чужие шуточки, формируются группировки «своих», а на заднем плане маячат люди, при взгляде на которых в голо-

ве пульсирует единственный вопрос: *«Мама дорогая, а это вообще кто и зачем их сюда пустили?»*. А учитывая, сколько жизни мы добровольно сжигаем на работе в процессе создания продукта, хочется не просто уныло пялиться в экран ноутбука, а хотя бы, чтоб было интересно тут находиться.

Мне в свое время крупно, просто сказочно повезло: в собственном проекте я умудрился получить абсолютный карт-бланш и сам лично решал, кого из кандидатов брать к себе в команду, а кого не брать. Звучит как фаундерская мечта из пабликов про внезапный успех, правда?

Только вот суровая производственная практика быстро засунула эту мечту в глубокое темное место. Выяснилась банальная вещь: абсолютная свобода выбора — это ни разу не гарантия коммерческого успеха и даже не страховка от убытков. Одно дело — собрать вокруг себя группу приятных, эрудированных собеседников для высокоинтеллектуальных разговоров по пятницам и распития крафтового пива. И совсем другое — запустить с этой интеллигентной, ранимой тусовкой жесткую боевую разработку, когда дедлайн закончился еще в прошлом квартале.

Перед нами мгновенно разворачиваются три основных задачи:

Заманить в проект ментально адекватных разработчиков. Не случайных людей с улицы и не амбициозных мечтателей из телеграм-чатов в духе *«ищу тиммейта для создания убийцы Uber на чистом энтузиазме»*.

Ввести их в реальный курс дела. Настроить онбординг так, чтобы новые сотрудники не сидели неделями, тупо пялясь в монитор с выражением лица *«Господи, за что мне эта жизнь?»*.

Синхронизировать вектор движения. Заставить всех гребти в одном направлении. В противном случае каждый потащит архитектуру в свою сторону, и вместо управляемой лодки у вас получится увлекательная гребля на месте.

Но это в общем. А в частности, мы ждём, что с новыми людьми должны приходить: компетенции, знания, желание работать, поучаствовать в новой для себя истории, и, в качестве бонуса, прицепом идут завышенные эго-ожидания, нереализованные амбиции, бессмысленные междоусобные конфликты, непреодолимое желание переписать весь проект с нуля по собственным лекалам и вечный, тотальный дефицит личной ответственности за результат. Именно в этой невидимой зоне и тонут перспективные ИТ-продукты.

Просто потому, что эффективная команда — это не аккуратный, стерильный список сотрудников. Это живой, дикий капризный, шумный и временами токсичный организм. И если ты не умеешь с ним работать — готовься встречать почти управляемый хаос.

У каждого наивного фаундера в голове крутится одна и та же фантазия: нанять элитных, автономных инженеров. Чтобы они зашли в проект, молча взяли твое кривое ТЗ, переварили его внутри своих гениальных черепных коробок и вы-

дали на выходе безупречный работающий софт. Желательно еще, чтобы у них был релевантный опыт именно в твоей узкой нише.

Настоящие, трушные специалисты не ищут работу на Хедхантере. Это работа с мылом в заднице бегают за ними по всему интернету, умоляя взять миллион рублей в секунду и ДМС со стоматологией. И этот факт напрочь ломает классическую менеджерскую парадигму. Забудьте про разбор тысячи резюме от отчаявшихся выпускников онлайн-курсов «Войти в айти за две недели». Тут нужен точечный хантинг и индивидуальное вылизывание контактов в Твиттере или Линкедине.

Более того — эти капризные твари выбирают, с кем им работать, а не наоборот. Ты не заманишь их ни раздутой зарплатой, ни опционами, ни приторной HR-блевотиной в духе «мы тут одна большая дружная семья» и «у нас уникальная ламповая атмосфера». Если твой стартап с порога выглядит как горящий филиал Кащенко, элитный инженер просто развернется и уйдет. Они не голодают. Они дико разборчивые. И никакие мировые кризисы, санкции, пандемии или биржевые качели этого не меняют. В эпоху турбулентности хорошие специалисты становятся только еще более высокомерными и капризными. И да, как ни странно, это нормальное состояние рынка.

Конечно, скорость формирования команды на старте — вопрос жизни и смерти. Но упаси тебя бог торопиться и хватать

первого попавшегося, который согласился на твои условия здесь и сейчас. В перспективе этот компромисс сожрет твой проект изнутри, превратив ПО в легаси-помойку.

Немного реальной статистики, от которой хочется выйти в окно

Давайте обратимся к цифрам из нашего личного производственного ада. Как-то раз мы отчаянно искали бэкенд-разработчика со специфическим стеком и прямыми руками. Внимание, следите за руками и за тем, как тает твой бюджет:

Скрининг: Отсмотрели 100 резюме, из которых 50% отправились в шредер сразу, потому что авторы путали Java с JavaScript.

Первая волна: Сделали 50 приглашений на первичный созвон. Половина кандидатов ушла в глухое радиомолчание, треть написала «я передумал», остальные сказали, что-то вроде: «У меня уже на руках оффер от Яндекса на триллион денег, чао».

Воронка: Кое-как назначили 20 собеседований. До технического и культурного интервью дожили единицы, остальные отвалились на этапе «расскажите о своем худшем фак апе» (видимо, худшим фак апом было само появление на этом созвоне).

Финал: С горем пополам сформировали и направили 3 оффера. И мы начинаем молиться, чтобы хотя бы один согласился нажать кнопку «Принять».

Итог: Молились мы хреново. Получили 2 официальных отказа и один классический игнор. Кандидаты просто испарились.

На весь этот ментальный мазохизм мы бесследно спустили около трех месяцев чистого времени. Три месяца — это тебе кажется много? Пха! Я лично приходил на проекты, где нужного специалиста горько и безуспешно искали по 6—9 месяцев. Почти год, чувак! Да-а-а, корпоративный трудоголизм иногда даёт свои незабываемые плоды.

Какой из этого вывод? Если у тебя есть кэш, не пытайся строить из себя великого рекрутера. Нахер внутренний HR-отдел, который умеет только постить мемы в корпоративный канал. Сразу нанимай нормальное, ИТ-кадровое агентство, у которого уже есть готовая, прогретая база контактов нужных тебе психопатов. Да, услуги агентства стоят дорого. Но скорость подбора и фильтрации кандидатов окупает эти траты в первые же недели, спасая твою нервную систему от неминуемого инфаркта.

Кровавые процессы

Итак, допустим, чудо произошло. Ты занес чемодан денег ИТ-агентству, вылизал профили нужных кандидатов и каким-то чудом решил задачу номер один — нанял классных инженеров. Проблема решена?

Прямо сейчас перед тобой в полный рост встает вторая, куда более кровавая задача: как организовать процессы так, чтобы эти высокооплачиваемые примадонны смогли взаимодействовать друг с другом, а не перерезали друг друга в первую же неделю спринта.

Ведь какая главная фишка у топовых специалистов? Правильно — каждый из них обладает абсолютно непогрешимой, единственно верной и отлитой в граните точкой зрения по любому общему вопросу. Каждый этот комнатный Наполеон точно знает, как правильно разворачивать инфраструктуру, какими паттернами писать код, как тестировать систему и в каком именно порядке строить продукт. Уступить коллеге в техническом споре или, упаси бог, согласиться на противоположное мнение для такого кадра соизмеримо с личным геополитическим поражением и клеймом позора на всю оставшуюся жизнь.

Кейс из палаты мер и весов. На одном из моих проектов устроили невинное, казалось бы, общее обсуждение: *«Коллеги, а давайте решим, как мы будем работать с Git и*

ветками». Первые двадцать минут всё шло чинно-благородно: каждый эксперт важно извергал в пространство свое уникальное видение рабочего процесса. Но затем кто-то неосторожно, слишком громко хмыкнул в ответ на аргументы оппонента. Завязался спор на повышенных тонах.

Итог? Полтора часа. Пятнадцать взрослых, дипломированных мужиков с совокупной зарплатой в несколько миллионов рублей в месяц просто орали друг на друга, пытаясь доказать, чей подход к ветвлению репозитория более каноничен. Полтора часа оплаченного бизнесом времени улетели в трубу просто потому, что это каждого участника не помещалось в комнату.

Стремление выбрать лучшее архитектурное решение из лучших, конечно, теоретически заслуживает уважения. Но давайте посмотрим на факты: ваш стартап с вероятностью в 90% закроется нахер уже через год из-за того, что у инвестора кончатся деньги. И вся эта ваша великая священная война за названия веток в Гитхабе не будет иметь вообще никакого значения в масштабах вселенной. Разве оно того стоит? Ну серьезно, ребята.

В таких условиях решать вопросы стандартными методами фасилитации и «демократического голосования» — это чистой воды утопия. Если дать этой банде свободу слова, они будут бесконечно рефлексировать, мериться авторитетами и топить проект в бесконечных дискуссиях.

Что делать? Правильный ответ — никто, блядь, не зна-

ет. Любой, кто продает вам на курсах «универсальную формулу построения кросс-функциональных команд», — обычный инфоцыган. Никакой серебряной пули не существует в природе. Всё индивидуально ровно в той степени, в какой люди уникальны и разнообразны в своих девиациях, а создание коммерческого ПО — это чистой воды творческий процесс, а не сборка табуреток на конвейере.

Тем не менее есть пара административных ломов, которыми можно вскрыть эту ситуацию и, как минимум, сэкономить время.

Вариант первый — классическая диктатура. На сцене материализуется сильный, авторитарный лидер, чей авторитет в команде безусловен и не оспаривается. Дальше события развиваются по двум сценариям. Либо вождь благосклонно выслушивает каждого придворного инженера и сам выбирает финальный вектор, либо он вообще никого не слушает и директивно спускает сверху приказ: что, как и в каком порядке делать. Все остальные беззвучно слушаются и повинуются.

Но спешу вас расстроить: оба сценария — так себе решения с точки зрения эффективности. Попытка играть в демократию и устраивать голосование большинства — это заведомо проигрышный, импотентский подход, потому что толпа (даже состоящая из сеньоров) никогда не гарантирует выбор технически верного решения. С другой стороны, если врубить глухого тирана, можно легко прохлопать позицию

какого-нибудь тихого интроверта, чье предложение в данный конкретный момент могло бы спасти ваш проект от касковой катастрофы.

Вариант второй — сеанс цивилизованной бюрократии. Это организованное совещание, где у тебя, наша куколка лучшего менеджера, внезапно есть жесткий регламент, вменяемый модератор, строгий порядок и план. Не так, как вы привыкли делать в своих стартапах: *«Ой, пацаны, тут созрел вопросик, какой мини-пылесос для клавиатуры купить в офис, давайте срочно созвонимся на полтора часа всей компанией и перетрем»*. Нет.

Нормальная процедура выглядит так:

— Заранее готовится четкий спектр вопросов и предлагаемые варианты решения.

— Весь этот материал принудительно рассылается команде с требованием ознакомиться *до* созвона, а не во время него.

— Команда собирается в Зуме, где модератор сразу задает жесткие правила игры и темп дискуссии.

— Высказываются строго по очереди. Любые попытки перебить оппонента, съехать на личности и превратить совещание в колхозный базар пресекаются модератором на корню.

— Организатор фиксирует мнения, собирает их в кучу, аргументированно поддерживает или разносит позицию каждого спикера отдельно, после чего единолично принима-

ет финальное решение.

При таком подходе ваши капризные инженеры хотя бы будут пребывать в иллюзии, что их экспертное мнение кому-то важно, а сама процедура обсуждения имеет человеческий вид и не вызывает желания немедленно обновить резюме на Хедхантере.

Старая притча в тему. Собрались как-то звери в лесу на стратегическую сессию. Сначала на трибуну выскочил заяц и высказал общее опасение. Затем поднялся осел и выразил глубокое общее недоумение. А в самом конце встал лев, лениво зевнул и озвучил общее мнение.

Как-то так это и должно работать в вашем проекте. Намек, думаю, предельно ясен.

Хотя, если честно, на дистанции всё это не имеет вообще никакого значения. Все ваши великие споры об архитектуре, или паттернах — это типичные административные терки о том, с какой стороны разбивать одно и тоже яйцо. Вы в любом случае напилите гигантскую кучу системных ошибок по примерам из этой книги и благополучно развалите проект.

Самое главное на старте — вовсе не найти идеальное решение, а сделать так, чтобы все согласились кодить по единому, пусть даже кривому стандарту, и наконец-то оставили попытки поубивать друг друга в рабочее время. Поэтому выбирайте проверенные временем подходы.

Элитный заповедник, или Зачем проекту пытливые умы

Итак, допустим, худшее позади. Твоя команда наконец-то действует организованно, без поножовщины и как единый, относительно вмняемый организм. Поздравляю.

В своей практике я всегда исходил из одной железобетонной парадигмы: нанимать нужно людей с исключительно **пытливым умом**. Это значит, что Homo Sapiens за клавиатурой всегда находится в состоянии перманентного, зудящего расширения своего кругозора. И абсолютно неважно, в какой сфере у него горит интерес: будь то природа квантовых полей, устройство ДНК, геополитические расклады Средневековья или тонкости перехода от строгого Ренессанса к безумному Барокко.

Зачем это нужно тебе, мой юный и доверчивый читатель? Какая бизнесу польза от того, что бэкендер по ночам читает труды Спинозы? Дело в том, что системное познание нового — это мышца. Если человек регулярно тренирует и развивает этот навык, он автоматически дарит твоему, пока ещё, перспективному стартапу три критически важные вещи, в которые тебе не придется вкладывать ни копейки ментального ресурса и контроля:

Первое. Стремительная утилизация незнания.

Инженер, умеющий и любящий жрать новую информа-

цию, при столкновении с нестандартной, неоднородной задачей не побежит плакать в чат. Он молча найдет нужный массив документации, за пару часов отсканирует его по диагонали, хладнокровно отбросит маркетинговую шелуху и освоит ровно тот кусок технологий, который необходим здесь и сейчас.

Второе. Дотошность как религия.

Хроническая привычка ковыряться до самых кишок в том, как устроены детали любого процесса, системы или явления, творит чудеса. Вероятность того, что в рамках твоей корявой задачи такой человек найдет, выберет и применит самый оптимальный, красивый и безопасный вариант решения, стремится к ста процентам. Причем он сделает это на чистом инженерном перфекционизме — даже если он уже мысленно свалил в другую компанию и это его последние три дня невнятного пребывания на твоём умирающем проекте. Он просто не умеет мусорить там, где работает.

Третье. Эффект самообучающегося улья.

Команда, состоящая из людей с пытливым умом, неизбежно превращает банальную курилку в элитарный интеллектуальный клуб. Они постоянно и совершенно бесплатно для твоего бюджета обмениваются профессиональным фидбэком и свежими знаниями. Они дискутируют об умных вещах и технологических трендах, попутно вытягивая за уши компетенции друг друга.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.