



Владимир Самородов

Системное мышление и бизнес анализ для руководителей и тех кто ими хочет стать

Владимир Самородов

**Системное мышление и бизнес
анализ для руководителей
и тех кто ими хочет стать**

«Автор»

2026

Самородов В. А.

Системное мышление и бизнес анализ для руководителей и тех кто ими хочет стать / В. А. Самородов — «Автор», 2026

Эта книга — для тех, кто управляет. И для тех, кто хочет управлять. Это практическое руководство по системному мышлению, написанное человеком, который проверил все принципы на реальных проектах. Без воды только инструменты, кейсы и решения, которые помогают превращать хаос в структуру. Вы узнаете:— Как перестать «тушить пожары» и начать проектировать систему.— Как проводить переговоры, когда на кону — бюджет, сроки и репутация.— Как управлять командой, когда каждый второй думает об уходе.— Как защитить проект от провала во время кризиса.— Как строить карьеру в эпоху неопределённости.— Как увеличить горизонт планирования до 10–15 лет.— Как принимать решения, когда нет правильного ответа. Для кого эта книга:— для руководителей всех уровней — от тимлидов до топ-менеджеров;— для основателей и владельцев бизнеса;— для тех, кто готовится к переходу на управленческую позицию;— для всех, кто устал от хаоса и хочет видеть систему там, где другие видят хаос.

© Самородов В. А., 2026

© Автор, 2026

Владимир Самородов

Системное мышление и бизнес анализ для руководителей и тех кто ими хочет стать

Аннотация



Меня зовут Владимир Самородов.

Я родился левшой в мире, где всё было сделано для правой руки. В 90-е родители купили мне велосипед «Эврика» — не такой, как у всех. Пока другие подростки катались на «Десне», я осваивал нестандартную механику и учился объяснять другим, почему моя техника — не брак, а преимущество.

Я тогда не задумывался об этом, но именно так работает системное мышление: когда ты понимаешь, как устроена конструкция, ты видишь её скрытые возможности. А когда ты видишь их — ты можешь объяснить другим, почему твой подход лучше. Даже если он не похож на всех.

Это был мой первый урок системного мышления.

Следующий урок мне преподавал Советский Союз. Тем, что просто исчез. Та стабильность системы, которая меня окружала, резко исчезла, и начали формироваться новые системы. В основном бизнес-модели. Тогда я и начал интересоваться ими. Для меня, как для ребёнка, привыкшего к миру правшей, перестраиваться было нормально. Но для взрослых это был шок, и это коснулось и меня.

Потом были HTML и Assembler — ещё до университета. А в университете я учился на инженера системотехника — углублённо изучал комбинаторику и прикладную теорию цифровых автоматов. И одновременно занимался скалолазанием и альпинизмом в сборной университета.

Затем 28 лет в IT — от кода до управления проектами с миллионными бюджетами. Строил системы с нуля, управлял командами от 5 до 450 человек, создавал бизнесы и выходил из них с прибылью. Я инвестировал в стартапы и менторствовал основателей — не потому, что я всё знаю, а потому что я знаю, где лежат грабли. Мои собственные ошибки — это лишь малая часть того, что я могу показать другим. Я целенаправленно учился у лучших: проходил

VIP-акселераторы, работал с Ренатой Джордж, впитывал опыт топ-экспертов. А главное — я научился видеть системные риски там, где другие видят удачу. Моя насмотренность позволяет не просто избегать ошибок, но и замечать скрытые угрозы, о которых никто не говорит. Именно это я передаю тем, кого менторю.

Я написал эту книгу не для теоретиков. Я написал её для тех, кто каждый день принимает решения в условиях неопределённости.

Внутри — никакой воды. Только инструменты, которые работают в реальных проектах: Как перестать «тушить пожары» и начать проектировать систему.

Как проводить переговоры, когда на кону — бюджет, сроки и репутация.

Как управлять командой, когда каждый второй думает об уходе.

Как защитить проект от провала во время M&A.

Как строить карьеру в эпоху кризисов и неопределённости.

Как увеличить горизонт планирования до 10–15 лет.

Как принимать решения, когда нет правильного ответа.

Для кого эта книга:

Для руководителей проектов, тимлидов, директоров по IT.

Для основателей стартапов и владельцев бизнеса.

Для аналитиков и PM, которые готовятся к переходу на управленческий уровень.

Для всех, кто хочет перестать быть исполнителем и начать управлять.

И помните: «Каждая система прекрасно спроектирована для того результата, который она выдаёт». Если результат не тот — меняйте систему.

Введение

Вы думаете, управление начинается с модных методологий, дорогих консультантов и сложных систем мотивации?

Нет. Оно начинается с умения видеть, как устроена система — и как её можно пересобрать, чтобы она работала лучше. Даже если никто вокруг не видит этого.

Меня зовут Владимир Самородов. Я родился левшой в мире, где всё было заточено под правую руку. В подростковом возрасте родители купили мне велосипед «Эврика» — не такой, как у всех. С рулём, который настраивался под тебя. С рамой, которая складывалась иначе. С багажником, который являлся частью несущей конструкции рамы. Пока 99% ребят катались на «Десне», я осваивал нестандартную механику. И учился объяснять другим, почему моя техника — не брак, а преимущество.

Ирония судьбы: главным доказательством этого преимущества стали девочки из нашего двора. Они быстро оценили, что именно на багажнике моего велосипеда можно безопасно кататься — в отличие от навесного багажника у «Десны», который не мог выдержать вес даже одного подростка и безбожно гнулся. Мой велосипед мой везти двоих — и даже троих, если мы ехали медленно. Это было не просто «удобно». Это было безопасно. Я тогда не задумывался об этом, но именно так работает системное мышление: когда ты понимаешь, как устроена конструкция, ты видишь её скрытые возможности. А когда ты видишь их — ты можешь объяснить другим, почему твой подход лучше.

Это был мой первый урок системного мышления.

Следующий урок мне преподавал Советский Союз. Тем, что просто исчез. Та стабильность системы, которая меня окружала, резко исчезла, и начали формироваться новые системы. В основном бизнес-модели. Тогда я и начал интересоваться ими. Для меня, как для ребёнка, привыкшего к миру правшей, перестраиваться было нормально. Но для взрослых это был шок, а это коснулось и меня.

Потом были HTML и Assembler — ещё до университета я изучил их по книгам без компьютера. А в университете я изучал комбинаторику и прикладную теорию цифровых автоматов. И одновременно занимался скалолазанием и альпинизмом в сборной университета. Там я увидел, как комбинаторика работает в реальности: скалы не дают тебе времени на раздумья, у тебя есть несколько комбинаций твоего тела с силой тяжести, трением и мышечными усилиями, — а цель не только выжить, но и подняться выше. Я получил второй спортивный разряд в забеге на скорость: 50 метров вверх по скале, обогнав кандидата в мастера спорта. Тогда я осознал когда ты на скале, мысли отключаются, мозг превращается в вычислитель, работающий на максимальной скорости. Там нет времени на сомнения. Есть только решение. Этот опыт оказался полезнее многих лекций: он научил меня принимать решения в условиях жёстких ограничений и с минимальным временем на размышление.

В университете у меня появился Pentium I 233 Mhz и наконец-то я смог практиковаться. Дальше — 28 лет в IT. Я прошёл путь от написания кода до управления проектами с бюджетами \$50M+ и командами до 450 человек. Строил архитектуру для корпораций и стартапов. Управлял продуктами и выводил их на рынок. Создавал бизнесы с нуля и выходил из них с прибылью. Инвестировал в стартапы и менторствовал основателей.

Но главное, что я понял за эти годы: любой бизнес — это проект. А управление проектами — это не про графики Ганта и схемы в той или иной модной нотации. Это про людей, переговоры, риски, решения и адаптацию.

Один из проектов, которым я руководил в роли исполнительного директора, работал в условиях жёсткого внешнего давления — компания попала в международные санкционные списки. Это потребовало от нас не просто технической перестройки, а полного пересмотра цепочек поставок, финансовой модели и команды. Этот опыт стал моей лучшей школой антикризисного управления и бизнес-анализа в реальном времени. Хорошо, что до этого у меня уже был опыт работы под санкционным давлением — но не в малом стартапе, а в крупных корпорациях с их запасом прочности и господдержкой. А вот тогда, в том стартапе, мы были одни против международной системы.

Именно тогда я окончательно убедился: системное мышление и навыки управления проектами — это универсальный язык, который работает в любой отрасли, любой стране и любой кризисной ситуации.

Эта книга — не мемуары. И не сборник теорий.

Это концентрированное практическое руководство по системному мышлению для руководителей, написанное человеком, который проверил все принципы на практике. В нём нет «вдохновляющих цитат» без реальной основы. Только инструменты, кейсы и решения, которые я использовал сам — и которые помогли сотням людей, которых я менторю.

Эта книга рабочий инструмент, который можно прочитать и сразу начать применять.

Кому эта книга?

Руководителям проектов, тимлидам, директорам по IT, которые переросли роль «исполнителя» и хотят управлять системно, а не «тушить пожары».

Основателям и владельцам бизнеса, которые строят компании и хотят превращать хаос в структуру.

Аналитикам, PM и будущим руководителям, которые готовятся к переходу на управленческий уровень.

Всем, кто устал от хаоса и хочет получить инструменты, чтобы управлять проектами, командами и карьерой.

Что вы найдёте в этой книге:

Главы 1–9 — фундамент: как мыслить системно, собирать требования, писать ТЗ, управлять проектами и командой.

Главы 10–13 — управление в сложных условиях: командообразование, M&A, защита от провала, переговоры.

Главы 14–16 — карьерный рост: как перестать быть исполнителем и стать руководителем, как мотивировать команду, как увеличивать горизонт планирования.

Глава 17 — антихрупкость: как строить карьеру в эпоху кризисов и неопределённости.

Глава 18 — системное мышление как главный управленческий инструмент.

Если вы готовы смотреть на систему не как на «чёрный ящик», а как на совокупность взаимосвязей, которые можно анализировать и менять — эта книга для вас.

И да, в ней будет велосипед. Но не только.

Глава 1. Системный сдвиг: как я перестал учить инструментам и начал учить мышлению «Я знаю, как работает каждый элемент. Но почему система всё равно не работает?»

Эту фразу я слышу от руководителей снова и снова. От директора по IT, который внедрил лучшие практики DevOps, но проекты всё равно срываются. От владельца бизнеса, который нанял топ-менеджеров, а компания не растёт. От тимлида, который обучил команду современным методологиям, а продукт всё равно не выходит в срок.

Они знают инструменты, процессы, регламенты. Они прочитали сотни книг, прошли десятки курсов. Но они не понимают одного: почему система не даёт нужного результата?

Все они пришли ко мне с одним и тем же запросом: «Я управляю отдельными частями, но не вижу целого».

Кто приходит ко мне за системным мышлением

Среди моих учеников — люди, которые уже достигли успеха в своей сфере, но столкнулись с потолком.

Руководитель IT-отдела в крупной корпорации. Он внедрил Agile, Kanban, DevOps. Команда работает, но бизнес всё равно недоволен. Почему? Потому что он управляет процессом, а не результатом.

Основатель стартапа. Продукт готов, команда собрана, инвестиции получены. Но компания не масштабируется. Он не видит, как его решения влияют на всю систему.

Топ-менеджер из госкорпорации. Он знает, как проходить тендеры, согласовывать бюджеты, отчитываться перед руководством. Но в условиях санкций и кризисов его привычные схемы перестают работать.

HR-директор. Она понимает людей, мотивацию, корпоративную культуру. Но её решения не всегда приводят к росту бизнеса. Почему? Потому что HR — это часть системы, а не сама система.

Финансовый директор. Он видит цифры, бюджеты, KPI. Но не всегда понимает, как его решения влияют на команду, продукт и клиентов.

Но есть ещё одна категория учеников, которых я выделяю отдельно. Это женщины, которые возвращаются в профессию после декрета. Они часто приходят с ощущением, что мир ушёл вперёд, а они остались. Что их опыт обесценился. Что они «выпали из обоймы».

Вот их истории.

Молодая мама двоих детей, муж — пенсионер. У неё было хорошее профильное образование, но больше десяти лет без работы — декреты, семья, быт. Когда она пришла ко мне, она не верила, что сможет вернуться. Через полгода работы в крупной телекоммуникационной компании она не просто вернула веру в себя — она закрыла два проекта, которые до неё столкнулись с сопротивлением. А через год выросла до руководителя и более 8 лет продолжает успешную трудовую деятельность.

Ещё одна мама двоих детей трудилась до этого в логистике и вела сложные коммуникации. Никакого управленческого бэкграунда. Сегодня она — прекрасный руководитель направления финансового анализа, и уже пять лет активно работает в телекоме. Её домен — логистика и коммуникации, но теперь она смотрит на неё через призму системного анализа.

Мама двоих детей с экономическим образованием, которая после семи лет жизни в деревне с детьми решила вернуться в профессию. Стала BI-аналитиком и активно трудится в дружном коллективе. Её аналитический склад ума, подкреплённый экономическим образованием, оказался идеальным фундаментом для работы с данными.

Мама троих детей, которая до декрета работала в консалтинге — не IT, а экономика. Сегодня она BI-аналитик и активно пишет SQL-запросы в Oracle и PostgreSQL. Её путь — пример того, как навыки работы с данными и аналитическое мышление можно перенести из одной сферы в другую.

Художник и мама. Более 15 лет она писала картины. Сегодня она — бесценный сотрудник направления UX/UI. Потому что дизайн — это та же композиция, те же цвета, та же гармония, хотя вместо холста — интерфейсы, а вместо красок — прототипы. Сейчас она управляет направлением UX/UI и руководит внутренними командами и внешними подрядчиками.

HR-подборщик персонала стала прекрасным аналитиком и ведёт проекты, а её основные заказчики — HR-директора. Она знает язык, на котором говорит бизнес, и теперь помогает ему принимать решения на основе данных.

Студентки из МАДИ — их больше пяти. Я преподавал им в университете. С математическим образованием они пришли в IT и теперь занимают позиции fullstack-аналитиков и менеджеров проектов. Их системное мышление, заложенное математикой, оказалось прочнее многих «профильных» IT-навыков и помогла им развиваться в управлении командами.

Основатели стартапов с абсолютно разным опытом — их объединяет страсть реализации своей идеи и отсутствие опыта в системного построения бизнеса.

Основатель финтех-стартапа. Блестящий технарь, 15 лет в разработке, написал код, который работал. Но он не понимал, почему инвесторы не хотят вкладываться, почему команда разваливается, почему продукт не взлетает. Я показал ему, что он управляет кодом, а не бизнесом. Через год он привлёк инвестиции, собрал управленческую команду и перестал быть «главным программистом».

Основатель IT-компании с 20-летним стажем. Всё знал, всё умел, но компания не росла. Он брался за всё сам — продажи, разработка, бюджеты, клиенты. Оказалось, что он не умеет делегировать и не видит, как его решения влияют на систему. Мы перестроили его подход: он перестал управлять людьми и начал управлять процессами. Компания выросла в 3 раза за два года.

Основатель стартапа в сфере логистики. У него было видение, но не было понимания, как выстроить процессы, чтобы команда работала без его участия. Он был «главным тушителем пожаров». Я научил его проектировать систему, которая работает без его постоянного вмешательства. Через полгода он ушёл в стратегию, а операционку передал команде.

Основатель EdTech-стартапа. Хороший продажник, слабый менеджер. Он нанимал людей, но они не задерживались. Я показал ему, что он нанимает «навыки», а надо нанимать «системных людей». Мы перестроили его подход к найму и управлению, и текучка сократилась в разы.

Основатель, который хотел «всё бросить». У него было три стартапа, все на грани закрытия. Он был вымотан, разочарован, готов был сдаться. Мы разобрали его систему: он распылялся, не фокусировался, не видел причинно-следственных связей. Мы определили его «точку невозврата» и перестроили приоритеты. Один стартап он закрыл, два — масштабировал. Он перестал быть «героем-одиночкой» и стал системным основателем.

И есть ещё один случай, который я вспоминаю особенно часто.

Это история нейроотличного студента, с которым в институте не могли общаться — его считали «сложным», «некоммуникабельным», «непонятным». Под моим началом он стал ведущим ETL-разработчиком в BI-отделе. И, что важнее, он научился правильно доносить свои мысли до оппонентов. Не потеряв себя, не сломавшись, а найдя язык, на котором его понимают.

Все эти люди — мои ученики. Они пришли из разных сфер, с разным бэкграундом, с разными страхами. Но всех их объединяло одно: им не хватало системного мышления. И когда они его получили — они перестали быть «винтиками». Они начали видеть систему.

История первая: велосипед, который сломал шаблон

В 90-е я катался на «Эврике». Не на «Десне», как 99% моих сверстников, а на велосипеде, который складывался не вбок, а сверху вниз. С рулём, который можно было настроить под себя. С багажником, который выдерживал больше, чем казалось.

Соседские мальчишки не понимали: «Это что за хрень? У тебя велосипед сломан?»

Я мог бы сказать: «Это просто другой велосипед». Но я учился объяснять, почему моя техника — не брак, а преимущество. Почему она устроена иначе. Почему она лучше подходит для моих задач.

Это был мой первый урок системного мышления: не суди по внешнему виду, разбирайся в том, как устроена система.

История вторая: компьютер без интернета

HTML и Assembler я изучал по книгам. Без компьютера. Без интернета. Просто сидел и представлял, как код превращается в работающую программу.

Я уже знал, что делать - когда у меня появился компьютер с Intel Pentium MMX 233 Mhz. Но главное, что я вынес из того периода: система работает не потому, что ты знаешь все кнопки. А потому, что ты понимаешь принципы.

Я не учил синтаксис. Я учился мыслить, как компьютер.

История третья: исчезновение страны, в которой я вырос

Следующий урок мне преподавал Советский Союз. Тем, что просто исчез.

Стабильность системы, которая меня окружала, рухнула за один день. Начали формироваться новые системы, в основном бизнес-модели. Тогда я и начал интересоваться ими всерьёз.

Для меня, как для ребёнка, привыкшего к миру правой, перестраиваться было нормально. Но я видел, как взрослые вокруг теряли ориентиры. Их системы рухнули, а новые они не готовы оказались строить.

Я понял тогда: системное мышление — это не про технологии. Это про способность видеть, как всё связано, когда привычные связи рвутся.

Чему я научился за 28 лет управления

Потом были университет, работа в госкорпорациях, стартапы, санкции, \$50M+ бюджеты, команды до 450 человек. Я строил архитектуру для Ростелекома и Росатома. Внедрял ERP, CRM, WMS, BI. Управлял продуктами и выводил их на рынок. Создавал бизнесы с нуля и выходил из них с прибылью. Инвестировал в стартапы и менторствовал основателей.

И каждый раз я видел одну и ту же картину: руководители тонут в инструментах, но не понимают систему.

Они знают, как внедрить Agile, но не знают, как он связан с бизнес-целями. Они слышали про OKR, но не понимают, как их связать с реальными результатами. Они управляют по KPI, но не видят, как эти KPI влияют на команду, продукт и клиентов.

Они знают, как «сделать», но не знают, «зачем». Они управляют процессами, но не видят системы.

Для кого эта книга

Эта книга — для вас, если вы узнаете себя в одном из этих портретов:

Руководитель IT-отдела в крупной компании. Он внедрил лучшие практики, но бизнес всё равно недоволен. Он знает, как управлять командой, но не знает, как управлять результатом. Хочет перестать быть «исполнителем» и стать «руководителем системы».

Директор по продукту. Он знает, как выводить продукты на рынок, но не знает, как выстраивать систему, которая делает это без его постоянного участия. Хочет масштабировать себя через системное мышление.

Основатель стартапа. Продукт готов, команда собрана, но компания не растёт. Он не видит, как его решения влияют на всю систему. Хочет перестать быть «главным героем» и стать «архитектором системы».

Топ-менеджер из госкорпорации. Он знает, как проходить тендеры, согласовывать бюджеты, отчитываться. Но в условиях санкций и кризисов его привычные схемы перестают работать. Хочет научиться принимать решения в условиях неопределённости.

Проектный менеджер. Он знает, как управлять задачами и сроками, но не знает, как управлять бюджетом, рисками и стейкхолдерами. Хочет перейти на следующий уровень.

Тимлид. Он знает, как вести команду, но не знает, как выстроить систему, в которой команда работает без его постоянного контроля. Хочет перестать быть «пожарным» и стать «архитектором».

Бизнес-аналитик, который хочет вырасти до руководителя. Он знает, как собирать требования, но не знает, как управлять проектами, командой и бюджетом. Хочет перейти от «исполнения» к «управлению».

Все они — мои ученики. Все они пришли ко мне с разным бэкграундом, разными целями и разными пробелами. Но всех их объединяло одно: им не хватало системного мышления.

Что вы найдёте в этой книге

Я не буду учить вас управлять по учебникам. Я не буду давать готовые рецепты «как внедрить Agile за 5 дней». Таких книг — сотни. И они не работают, потому что они учат инструментам, а не мышлению.

Я научу вас видеть систему.

Я покажу вам, как:

Разбирать любую систему на составляющие и понимать, как они связаны.

Переходить от хаоса к структуре.

Принимать решения, основанные на понимании, а не на интуиции.

Управлять проектами, командами и бизнесом так, чтобы они работали как часы.

Строить карьеру, устойчивую к кризисам.

Я дам вам инструменты, которые проверены на реальных проектах: от стартапов до госкорпораций. Я покажу ошибки, которые совершал сам, чтобы вы могли их избежать.

И последнее: почему я написал эту книгу

Потому что устал смотреть, как умные, талантливые люди топчутся на месте, не понимая, куда двигаться дальше. Потому что верю: системное мышление — это навык, который можно натренировать. Потому что хочу, чтобы вы прошли этот путь быстрее и дешевле, чем я.

И да, в этой книге будет велосипед. Но не только.

Глава 2. Методология — не догма, а каркас. Как я перестал быть формалистом и стал прагматиком

«Мы взяли Scrum, но он у нас не работает».

Эту фразу я слышал от десятков руководителей. Они честно пытались. Проводили ежедневные митинги, ставили спринты, делали ретроспективы. А результат был тот же: хаос, срывы сроков, уставшие команды.

Потому что они копировали методологию, не думая о контексте.

История первая: как я десять лет не пользовался ГОСТом

До 2015 года я прекрасно знал, что такое ГОСТ 34. Но я сознательно его не применял. Почему? Потому что он убивал скорость.

Я работал в компаниях, где важнее был результат, а не документ. Где клиенту нужно было решение, а не 50 страниц технического задания. Где команда ценила гибкость, а не формальности.

И это работало. Мы выпускали релизы быстрее конкурентов. Мы быстрее реагировали на изменения. Мы не тратили время на то, что не приносило ценности.

Я знал, что ГОСТ существует и пользовался им ранее. Но я сознательно выбирал не использовать его, потому что в моём контексте он был тормозом.

История вторая: 2015 год — «Алле, надо по ГОСТу»

В 2015 году компанию, в которой я работал, поглотила госкорпорация. Всё изменилось очень быстро.

— Владимир, — говорит мне специалист отдела закупки, — нам нужно техзадание по ГОСТу. Иначе тендер не пройдёт.

— Окей, — говорю, — будет вам ГОСТ.

Я открываю свежий ГОСТ 34 и вижу: там масса разделов, которые нужно заполнить. Требования, архитектурные решения, сценарии тестирования, планы внедрения, регламенты сопровождения...

И всё это нужно было делать по правилам, которые не учитывали гибкость.

История третья: как я скрещивал ГОСТ с гибкими методологиями

Я не мог отказаться от ГОСТа — тендер бы не прошёл. Но я не мог и просто копировать его — проект бы застрял. Нужно было найти баланс.

Тогда я начал адаптировать ГОСТ под гибкие методологии.

Где-то я добавлял текст, чтобы соответствовать требованиям закупщиков. Например, заполнял разделы про архитектурные решения и сценарии тестирования.

Где-то я убирал текст, чтобы сохранить гибкость. Например, сокращал объём документа до 5–10 страниц вместо 50. Убирал дублирующие разделы. Заменял сложные формулировки на простые.

Где-то я менял форму, сохраняя суть. Например, вместо 100-страничного ТЗ я делал 5-страничный документ с чёткими требованиями, а всё остальное — в приложениях.

Я не ломал ГОСТ, я его пересобираю под свои задачи. Сохранял структуру, но делал её живой. Добавлял разделы, которые были нужны команде. Убирал то, что не приносило ценности.

В итоге все были довольны:

Закупщики получили документ, который проходил тендеры.

Исполнители не утонули в бюрократии, и мы могли их контролировать для быстрого выпуска релизов.

Я получил рабочий инструмент, который не убивал гибкость.

Почему методологии не работают «из коробки»

У меня есть простое правило: методология — это каркас, а не стена. Она даёт тебе опору, но не диктует, куда идти.

Почему люди терпят неудачу?

Они копируют без адаптации. Прочитали книгу, посмотрели курс — и внедряют один в один. Но у них другой бизнес, другая команда, другие риски.

Они игнорируют культуру. Методологии создаются под определённые ценности. Если у вас культура «сделай быстро», а вы внедряете «сделай по документации» — будет конфликт.

Они забывают про людей. Команда не любит, когда ей навязывают процессы, в которых она не участвовала. Люди хотят понимать «зачем». Если они не видят смысла, они сопротивляются.

И ещё один важный момент: иногда методологию нужно не внедрять, а сознательно откладывать. Как я делал с ГОСТом до 2015 года. Это тоже выбор. И это тоже системное мышление.

Как я адаптирую методологии (практический алгоритм)

За годы работы я выработал систему. Она помогает мне не копировать, а адаптировать любую методологию под конкретный проект.

Шаг 1. Понять контекст

Я не беру методологию в руки, пока не отвечу на вопросы:

Что за бизнес? (стартап, корпорация, госкомпания)

Какая команда? (опыт, мотивация, состав)

Какие стратегические цели компании?

Какие SLA у компании перед клиентами?

Какие SLO между подразделениями?

В каком сегменте финансовой структуры компании я нахожусь (привет ЦФО)?

Какая техническая зрелость моего заказчика?

Без ответа на эти вопросы методология — просто набор букв.

Шаг 2. Взять из методологии только то, что даёт ценность

Я не внедряю методологию целиком. Я использую инструменты, которые решают реальные проблемы.

Если проблема — хаос в требованиях → беру из ГОСТа рекомендации по описанию требований.

Если проблема — медленная обратная связь → беру из Scrum ежедневные митинги.

Если проблема — непонимание архитектуры → беру из ArchiMate моделирование или более простую визуализацию.

Всё остальное — отбрасываю без сожаления.

Шаг 3. Адаптировать под свою скорость

В стартапе я не могу делать двухнедельные спринты и ежемесячные отчёты. Я делаю недельные спринты и отчёты по запросу. В госкорпорации и среднем бизнесе могут быть спринты 2–4 недельные, и это нормально для их скорости и бюрократии.

Скорость диктует формат. Не наоборот.

Шаг 4. Часто я принимаю решение единолично, но объясняю команде «зачем»

Я не провожу голосования. Я не собираю команду на совещание для обсуждения, какую методологию нам взять.

Я принимаю решение сам.

Потому что:

Я вижу общую картину, которую команда не всегда видит.

Я отвечаю за бюджет, сроки и результат.

У меня есть опыт, который команда не имеет.

Но есть одно условие: я всегда предварительно собираю информацию для взвешенного решения и в этом мне помогает команда и особенно эксперты и тимлиды. А затем объясняю команде, почему я принял именно это решение.

«Мы переходим на недельные спринты, потому что у нас нестабильная среда, и двухнедельный спринт — это слишком большой риск.»

«Мы будем использовать ГОСТ 34 только для архитектурных решений и требований. Всё остальное — упрощаем, чтобы не тормозить разработку.»

Шаг 5. Пересматривать процесс регулярно

Методология не статична. Я пересматриваю её эпизодически и в зависимости от размера компании, иногда раз в квартал, а иногда раз в год. Что работает — оставляем. Что нет — меняем.

Следует адаптировать методологии под ваш контекст. Не копировать, а брать лучшее и подстраивать под себя.

И последнее: почему это важно

Потому что мир не будет ждать, пока вы освоите идеальную методологию. Бизнес меняется, команды меняются, технологии меняются. Единственное, что остаётся неизменным — это ваше умение адаптироваться.

Методология — не догма. Это каркас, на котором вы строите свой дом. Дом должен быть удобным для вас, а не соответствовать картинке из журнала.

Иногда вы сознательно отказываетесь от методологии, потому что она мешает. Иногда вы адаптируете её под жёсткие требования заказчика. Иногда вы создаёте свой собственный подход из кусочков разных систем.

Главное — не быть формалистом. Относитесь к проекту как реалист и прагматик.

Глава 3. Ваш прошлый опыт — это не багаж, а фундамент

«То, что мы называем опытом, часто является просто памятью о наших ошибках» - Нассим Талеб

Одну фразу я слышу на каждой второй консультации: «У меня нет профильного образования, я ничего не умею».

Директор по производству, который 15 лет управлял заводом, думает, что его опыт не пригодится в ИТ. Руководитель отдела продаж, который строил команды и выводил продукты на рынок, считает, что его навыки не имеют отношения к управлению ИТ-проектами. Топ-менеджер из логистики уверен, что его знания о процессах перевозок бесполезны в ИТ.

Я всегда отвечаю одно и то же: давайте посмотрим на это с другого ракурса.

История первая: как директор по производству стал ИТ-руководителем

Сергей управлял заводом 15 лет. Он знал, как выстроить производственные процессы, как управлять людьми, как работать с бюджетом. Но когда его компания начала цифровую трансформацию, он почувствовал, что отстаёт.

— Я не знаю ИТ, — сказал он. — Я не понимаю, как управлять разработкой, как работать с Agile, как оценивать сроки.

— А что ты делал на заводе? — спросил я.

— Управлял производством. Отвечал за то, чтобы продукт был выпущен в срок, в нужном количестве и с нужным качеством. Управлял бригадами, складами, закупками.

— А ИТ-проект — это то же самое, — сказал я. — Только вместо станков — разработчики. Вместо складов — репозитории. Вместо закупок — бюджетирование. Суть та же: ты должен организовать процесс так, чтобы результат был достигнут с минимальными потерями.

Сергей задумался. А потом начал перекладывать свой опыт на новую область:

Его знание производственных процессов → понимание жизненного цикла разработки.

Его опыт управления бригадами → управление командами разработки.

Его понимание бюджетирования → управление ИТ-бюджетами.

Его навыки работы с поставщиками → управление внешними подрядчиками.

Через 6 месяцев он стал руководителем ИТ-направления в своей компании. Его заводской опыт оказался ценнее, чем он думал. Он знал, как организовать процесс, как управлять людьми, как работать с бюджетом. Оставалось только добавить инструменты.

История вторая: как продажник стал операционным директором

Анна работала в продажах 12 лет. Она строила команды, выводила продукты на рынок, управляла бюджетами. Когда ей предложили стать операционным директором в ИТ-компании, она испугалась.

— Я не понимаю, как управлять разработкой, — сказала она. — Я не знаю, что такое CI/CD, DevOps, микросервисы.

— А что ты умеешь? — спросил я.

— Строить команды, понимать клиентов, управлять деньгами, договариваться, видеть рыночные тренды.

— А операционный директор — это про то же самое, — сказал я. — Ты управляешь людьми, деньгами и процессами. Технологии — это просто инструмент, который помогает тебе делать твою работу быстрее и дешевле.

Сегодня Анна — успешный операционный директор. Она говорит:

«Мой опыт в продажах помогает мне каждый день. Я понимаю клиентов. Я понимаю команду. Я понимаю, как устроен бизнес. Технологии я учу по мере необходимости. Главное — я знаю, как управлять».

История третья: как начальник отдела логистики стал СТО

Михаил управлял логистикой в крупной компании. Он знал, как выстраивать цепочки поставок, как управлять рисками, как работать с подрядчиками. Когда компания решила создать IT-департамент, ему предложили его возглавить.

— Я не программист, — сказал он. — Я не могу быть CIO.

— А кто сказал, что CIO должен уметь программировать? — спросил я.

— Ну... я думал...

— CIO — это про управление IT-системами, — сказал я. — Ты умеешь управлять логистическими цепочками. Теперь будешь управлять IT-системами. Разница — в контексте. Но суть та же: ты видишь картину целиком, понимаешь, как связаны элементы, умеешь управлять рисками.

Михаил возглавил IT-департамент. Он не пишет код, но он выстроил процессы, которые позволяют компании работать с IT как с единой системой. И он делает это лучше, чем многие «технари», потому что он видит бизнес целиком.

Что такое домен и почему он важен для руководителя

У каждого из нас есть опыт в определённой сфере — «домене» (на языке DDD). У Сергея — производство. У Анны — продажи. У Михаила — логистика.

Когда вы переходите на управленческий уровень, вы не перестаёте быть экспертом в своём домене. Вы просто меняете инструменты и добавляете новые компетенции.

Ваш доменный опыт — это ваш фундамент:

Если вы работали в производстве, вы понимаете жизненный цикл продукта, качество, безопасность.

Если вы работали в продажах, вы понимаете клиентов, команды, бюджеты.

Если вы работали в логистике, вы понимаете, как строить системы управления запасами, маршрутами, сроками.

Если вы работали в банке, вы понимаете, как работают транзакции, скоринг, риски.

Если вы работали в образовании, вы понимаете, как выстраивать обучение и обратную связь.

Вы не начинаете с нуля. Вы начинаете со знанием предметной области.

И это ваше преимущество. Потому что работодатели ищут не просто «IT-руководителей». Они ищут руководителей, которые понимают их бизнес.

Как использовать свой прошлый опыт (практический алгоритм)

Когда ко мне приходит новый ученик, я задаю ему один вопрос:

«Что ты уже умеешь, что можно применить в новой роли?»

И мы вместе разбираем его прошлый опыт.

Шаг 1. Выпиши свой предыдущий опыт

Что ты делал на прошлой работе? Какие задачи решал? С какими людьми взаимодействовал? Какие результаты были?

Шаг 2. Найди аналогии в целевой IT-роли

Если ты управлял людьми → ты сможешь управлять командами.

Если ты работал с бюджетами → ты сможешь управлять IT-бюджетами.

Если ты взаимодействовал с клиентами → ты сможешь управлять требованиями.

Если ты отвечал за результат → ты сможешь управлять проектами.

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.