

СОДЕРЖАНИЕ

ОТ АВТОРА	5
БЛАГОДАРНОСТИ	6
КТО ТАКОЙ БИЗНЕС-АНАЛИТИК?	7
ОСНОВЫ РАЗРАБОТКИ ПО	9
Роли в команде	9
Менеджер проекта (Project manager)	9
Менеджер продукта (Product manager)	9
Архитектор (Architect)	10
Аналитик (Analyst)	11
Дизайнер (Designer)	12
Специалист по информационной безопасности (Application security)	13
Разработчик (Developer)	13
Тестировщик (QA Engineer)	14
Техническая поддержка (Tech Support)	14
Жизненный цикл ПО	15
Жизненный цикл проекта	16
Этап планирования	18
Этап анализа требований	20
Этап проектирования	20
Этап разработки	20
Этап тестирования	20
Этап внедрения и поддержки	21
Стандарты жизненного цикла ПО	21
Модели разработки ПО	26
Классические модели разработки	27
Гибкие модели управления командой Agile и Kanban	34
Гибкие модели управления командой Scrum	51
ОСНОВЫ РАЗРАБОТКИ ТРЕБОВАНИЙ	65
Заинтересованные лица	65
Типы заинтересованных лиц	67
Роли заинтересованных лиц	68
Профили заинтересованных лиц	69
Приоритеты проекта	70
Пользователи	71
Классы пользователей	71
Выявление классов пользователей	73
Для чего нужна классификация пользователей	73
Организационная модель	74
Требования	78

СБОР ТРЕБОВАНИЙ	83
Эффективность различных каналов коммуникации	84
Интервьюирование	85
Этап 1. Подготовка	85
Этап 2. Проведение встречи	86
Этап 3. Действия после проведения интервью	87
АНАЛИЗ ТРЕБОВАНИЙ	88
Выделение пользовательских историй в отдельные пакеты	88
Описание требований с помощью вариантов использования	89
Нормальное направление варианта использования	91
Альтернативные направления варианта использования	92
Исключения вариантов использования	93
Последовательность вариантов использования	93
Выявление вариантов использования	94
Преимущества применения вариантов использования	95
Риски и ошибки при применении вариантов использования	96
Описание требований с помощью процессного подхода	97
Основные понятия процессного подхода	98
Способы описания бизнес-процессов	100
Виды бизнес-процессов	101
Нотации для моделирования бизнес-процессов	104
Структурные модели — нотация IDEF	105
Модели потоков работ — нотация eEPC	112
Модели исполняемых процессов — нотация BPMN	134
Модели потоков данных — нотация DFD	176
Модели потоков данных — нотация UML	179
ДОКУМЕНТИРОВАНИЕ ТРЕБОВАНИЙ	242
Документ об образе и границах проекта (Product Vision)	242
Информация, которую должен содержать документ	
об образе и границах проекта	243
План управления требованиями (ПУТ) и план управления документами (ПУД)	245
Документирование вариантов использования	246
Спецификация требований к ПО	249
Сколько спецификаций необходимо	251
Спецификация требований в проектах гибкой разработки	261
ПРОВЕРКА ТРЕБОВАНИЙ	264
Типовые ожидания от требований	264
Методы обеспечения качества требований	267
Трассировка требований	268
СОГЛАСОВАНИЕ ТРЕБОВАНИЙ	269
Три золотых правила согласования	269
Зоны ответственности подразделений	270
Условия успешной реализации процедуры согласования	270
Алгоритм и способы согласования	271
Оформление результатов согласования	272
Пример оформления согласования требований	274
ПРИЛОЖЕНИЯ	275
IT-сленг	275
Литература по бизнес-анализу	285
ОБ АВТОРЕ	287

ОТ АВТОРА

В книгах вроде «*Название профессии*» пособие для начинающих» обычно много написано о том, как долго автор к ней шел, сколько статей написал, в каком университете преподавал и т. д. Так вот, ничего подобного у меня не было. Два года назад я не только не планировал, что буду писать эту книгу, но даже и не думал об этом. Но так получилось, что ко мне обратились две очаровательные девушки с просьбой научить их бизнес-анализу, поскольку на тот момент я был старшим бизнес-аналитиком нефтяной компании «Роснефть» и имел большой опыт работы в профессии в российских и зарубежной компаниях. По итогу я все зафейлил, собрав все преподавательские грабли, какие только было можно. Выяснилось, что нормальной литературы по профессии просто нет: она или в общих чертах рассказывает о том, чем занимаются бизнес-аналитики, какими они бывают и т. п., или рассчитана на тех, кто уже работает в этой области и понимает, что конкретно изучает этот специалист. Девушки, которые попросили меня научить их бизнес-анализу, были не только очаровательными, они оказались еще и крутыми преподавателями в своих сферах, причем со знанием и опытом не только из отечественного высшего образования. Поэтому родилась мысль исправить эту ситуацию, написав книгу и создав курс «Бизнес-аналитик». За время написания я успел побывать руководителем команд, которые разрабатывали и сопровождали несколько информационных систем федерального значения, и запустить собственный стартап в сфере финтеха. Это позволило добавить в книгу информацию по hard и soft skills, которые я ожидал увидеть от кандидатов на позицию бизнес-аналитика уровня джуниор. По возможности развернуто я разобрал проблемы, возникающие у новых сотрудников при вводе их в должности.

Предупрежу: если кто-то думает, что после прочтения этой книги сможет стать бизнес-аналитиком и устроиться на работу за много-много денег, то я его разочарую, потому как придется прочесть и изучить еще немало материала. Однако моя книга станет тем скелетом, на который, как мясо, будут нанизываться новые знания, что в итоге позволит вам вырасти в крепкого специалиста.

БЛАГОДАРНОСТИ

В процессе написания этой книги о профессии бизнес-аналитика я осознал, насколько важна поддержка и помощь многих людей, без которых этот проект не стал бы реальностью. Хочу выразить свою искреннюю благодарность всем, кто способствовал созданию этой работы.

Прежде всего я благодарен своим коллегам и бизнес-партнерам, которые делились знаниями и опытом в области бизнес-анализа. Ваши ценные советы и конструктивная критика помогли мне глубже понять сложные концепции и сделать материал более доступным для читателей.

Я также хочу поблагодарить экспертов в области бизнес-аналитики, с которыми мне посчастливилось пообщаться. Ваши инсайты и рекомендации обогатили содержание книги и помогли мне взглянуть на предмет с разных точек зрения. Надеюсь, что читатели оценят тот опыт, которым вы щедро поделились со мной.

Особую благодарность хочу выразить Ирине Пинхасик. Ее бэкграунд руководителя проектов в сфере цифрового образования в различных европейских странах позволил мне использовать в книге современные мировые тенденции образования. Это должно сделать процесс усвоения информации для читателя наиболее комфортным и продуктивным.

Огромная благодарность команде редакторов и рецензентов, которые помогли улучшить текст и сделать его более структурированным. Ваш профессионализм и внимание к деталям сыграли ключевую роль в создании качественного продукта. Я ценю вашу работу и усилия, вложенные в этот проект.

В заключение хочу поблагодарить всех тех, кто был частью этого пути. Каждое ваше слово поддержки, каждая идея и каждый совет сделали эту книгу лучше. Надеюсь, что она станет источником вдохновения и практических знаний для многих.

С уважением, *Валерий Комсков*

КТО ТАКОЙ БИЗНЕС-АНАЛИТИК?

«...если бы среди философов установилось согласие относительно значения слов, то почти все их споры были бы прекращены»

Рене Декарт, XVII век

На самом деле вопрос не такой простой, каким кажется. Надеюсь, ни для кого не секрет, что «бизнес-анализ», «анализ бизнеса» (аудит) и «бизнес-аналитика» — это совершенно разные понятия и кроме «похожих слов» между ними не так много общего?

Но даже если мы возьмем конкретно бизнес-аналитиков, то выяснится, что есть два их вида:

- обычный (не IT) бизнес-аналитик — тот, кто работает в контексте бизнеса в целом. Такой специалист участвует в совершенствовании процессов, оптимизации издержек и т. д.;
- бизнес-аналитик в IT — тот, чья работа происходит в рамках отдельных IT-проектов — закупки, внедрения, внесение изменений и т. д. Его задача — взаимодействие с заинтересованными лицами со стороны заказчика (бизнес-пользователи, технический персонал и т. д.) для сбора требований, исходящих от бизнеса.

При этом IT — крайне снобистская сфера, и за пределами крупных корпораций о бизнес-аналитиках не из IT обычно никто не знает.

В то же время в отрасли нет четкого деления на «бизнес-» и «системных» аналитиков. В каждой компании (а подчас и в отдельных командах внутри одной компании) границы их компетенций сильно размыты, вплоть до того, что зачастую на проекте вообще нет системного аналитика, а его обязанности распределены между бизнес-аналитиком и разработчиками. И это не российская особенность: в мире также нет единого однозначного определения, достаточно почитать формулировку из библии бизнес-аналитиков BABOK.

Я сознательно крайне путано написал эту главу, чтобы вы увидели атмосферу, в которой вам придется работать. Бизнес-аналитик — это не та профессия, в которой

Кто такой бизнес-аналитик?

четко сказано, что «нужно копать отсюда и досюда». Вы постоянно будете сталкиваться с требованиями, которые зачастую противоречат не только друг другу, но и здравому смыслу, заказчики сами не будут знать, чего хотят, а вы нередко не будете знать, осуществимы ли в реальности их желания. Но вместе с тем бизнес-аналитик — одна из самых «инженерных» профессий, ибо все остальные члены команды реализуют то, что вы придумали и написали/нарисовали.

Говорят, девизом IBM в стародавние времена было утверждение: *«мы продаем нашим заказчикам не то, что они хотят, а то, что им нужно»*. Уж не знаю, насколько это правда, но формулировка весьма показательна. Со временем, когда вы наберетесь опыта, вам нужно будет следовать этому девизу.

Атмосферная часть закончилась, далее информация в книге будет гораздо более структурированной. Ну а чтобы вы знали, что ответить на собеседовании на этот вопрос, просто выучите определение из ВАВОК, который не просто так называется «Руководство к своду знаний по бизнес-анализу»®.

ОСНОВЫ РАЗРАБОТКИ ПО

Роли в команде

Менеджер проекта (Project manager)

Проект — это временное предприятие, направленное на создание уникального продукта, услуги или результата.

Менеджер проекта — руководитель проектной команды, ответственный за управление проектом, достижение цели проекта в рамках бюджета, в срок и с заданным уровнем качества.

Менеджер релизов — в реалиях российских Enterprise-компаний так называются руководители команд, управляющие процессом, когда IT-продукт из проекта уже превратился в сервис, но находится на доработке согласно запросам на изменения (ЗИ).

Менеджер продукта (Product manager)

Продукт — конечный осязаемый результат (продукт, сервис, услуга), предоставляемый клиенту. Менеджер продукта задает стратегию развития продукта. Он знает все о желаниях потребителя, предугадывает, что будет с рынком (проводит анализ рынка) и т. п., хорошо представляет продукт с точки зрения потребителя, но не погружается в технический контекст.

Эти роли различаются тем, что product manager придумывает новые функции, которые можно добавить в продукт, а project manager ставит задачу подчиненным реализовать эти новые функции и сообщает о сроках, когда они будут готовы. Product manager решает, что сделать, project manager — когда и как.

Архитектор (Architect)

Архитектура ИС — это набор структурных элементов, организованных определенными способами для взаимодействия друг с другом, которые складываются в единый программно-аппаратный комплекс. Он выстроен для достижения конкретных бизнес-целей.

Архитектура — понятие глобальное, его можно разделить на несколько частей, переходя от общего к частному. В работе аналитика встречаются следующие виды «архитектур».

Архитектура предприятия / корпоративная архитектура (Enterprise Architecture) — высокоуровневая архитектура всего предприятия, покрывающая бизнес-потребности ИТ-способностями. Корпоративная архитектура фокусируется на определении потоков и бизнес-процессов, действий, функций, информации, данных и технологий предприятия, а также на вызовах, стоящих перед ИТ и необходимых для того, чтобы эффективно применить технологию в ответ на изменение бизнес-потребностей.

Бизнес-архитектура (Business Architecture) описывает все бизнес-процессы, бизнес-акторы, бизнес-сущности и бизнес-правила с точки зрения бизнеса. Бизнес-архитектура не зависит от применяемых в разработке технологий.

Информационная архитектура (Information Architecture) определяет структуры данных и описывает все потоки данных, которые используются для поддержки бизнес-архитектуры. Такие операции, как идентификация, систематизация, категоризация, хранение данных, относятся к информационной архитектуре. Может представляться в виде модели данных (Data Model).

Архитектура решения (Solution Architecture) — архитектура программного обеспечения (ПО), которое реализует функции бизнес-архитектуры.

Технологическая архитектура (Technology Architecture) описывает архитектуру ИТ-окружения, которое используется для поддержки информационной архитектуры и архитектуры решения.

Системная архитектура (System Architecture) — представление системы, которое показывает реализацию функциональных возможностей аппаратными средствами и компонентами ПО, устанавливает связь архитектуры программного обеспечения и архитектуры аппаратных средств, а также регламентирует взаимодействие пользователя с этими компонентами.

Архитектура ПО (Application Architecture) — составная часть системной архитектуры. Описывает организацию системы с точки зрения программных компонентов, из которых она состоит, и связи между компонентами.

Архитектура данных (Data Architecture) — составная часть системной архитектуры. Описывает структуры данных и логические связи между ними.

Каждым из видов архитектуры занимается соответствующий архитектор.

Бизнес-архитектор (Enterprise Architect) отвечает за то, чтобы бизнес-стратегия компании использовала правильную архитектуру технологических систем для достижения своих целей. ЕА несет огромную ответственность и, как правило, подчиняются непосредственно директору по информационным технологиям (CIO). Они должны идти в ногу с последними тенденциями в технологиях и определять, подходят ли технологии для компании. ЕА берет бизнес-стратегию компании и описывает архитектуру технологических систем, которая будет необходима для поддержания этой стратегии. Архитекторы уровня enterprise ориентируются на бизнес-потребности, придумывают, как решить какую-либо бизнес-задачу, разрабатывают глобальный план работы приложения и алгоритм его взаимодействия с другими. ЕА должен представлять, как все приложения работают вместе, иметь всю информацию. Он принимает концептуальные решения.

Архитектор решений (Solution Architect). Если ЕА решает, что делать, то SA — как делать.

Архитектор решений отвечает за разработку одного или нескольких приложений или сервисов в организации. Решает проблему, которую определил бизнес:

- как технологии могут быть использованы для решения бизнес-проблемы;
- какую платформу или стек технологий можно использовать для создания решения;
- как станет выглядеть приложение, какими окажутся модули и как они начнут взаимодействовать друг с другом;
- как вещи будут масштабироваться и поддерживаться.

Архитектор решений часто работает под методическим руководством бизнес-архитектора.

Technical (software) Architect — техлид в обычных реалиях. Это технический эксперт, который пытается решать проблемы, поставленные SA-архитектором. ТА ставит задачи разработчикам и занимается уровнем реализации конкретной системы. На уровне конкретной информационной системы определяет стек технологий этой системы, сервисы, подходы к разработке, выбирает БД и интерфейсы для взаимодействий с другими системами.

Аналитик (Analyst)

Бизнес-аналитик (Business Analyst) описывает бизнес-процессы, как они есть (as is), и, общаясь с заказчиком и заинтересованными сторонами, составляет

предложение, какими они должны быть (to be), чтобы решить задачи бизнеса (достичь целей).

Бизнес-аналитик фиксирует бизнес-требования и их решения по реализации в виде бизнес-процессов. При этом может рисовать схемы бизнес-процессов, писать use case — бизнес-сценарии, которые необходимо проверять при приемке решения. Бизнес-аналитик выбирает любой способ фиксирования результатов своей деятельности и передает эту информацию системным аналитикам каждой из систем, которые участвуют в реализации решения. Они составляют свои документы.

Системный аналитик (Systems Analyst) становится связующим звеном между внедрением нового продукта, то есть усовершенствованием бизнес-процессов, и его разработкой. Он опрашивает заказчика, занимается сбором функциональных требований, фиксирует их, обсуждает со своей командой разработки, после чего команда принимает совместное решение о том, какую конфигурацию внедрения программного продукта предложить заказчику. После обсуждения и согласования аналитик описывает постановку задачи в документе, который согласовывает с заказчиком.

Продуктовый аналитик (Product Analyst) — мостик между бизнесом и данными. Он работает рука об руку с менеджером продукта и помогает продуктовой команде принимать верные решения.

Продуктовая аналитика позволяет увидеть, как пользователи взаимодействуют с продуктом. Условно можно выделить две задачи продуктовой аналитики: сбор данных и их интерпретация.

Сначала продуктовый аналитик собирает множество цифр из разных источников: какие кнопки нажимают пользователи, как часто они используют продукт, какие функции продукта популярны, а какие нет. Эти измерения показывают, что происходит с продуктом, но не объясняют почему.

На втором этапе аналитик вытаскивает из цифр инсайды, которые проливают свет на поведение пользователей. Благодаря этому продуктовая команда понимает, какой продукт она сделала и куда двигаться дальше.

Таким образом, продуктовый аналитик работает с данными, изучает метрики, строит воронки и следит за тем, к каким результатам приводит малейшее изменение продукта.

Дизайнер (Designer)

User Experience (UX) — это путь клиента от точки входа до точки выхода. То, насколько клиенту удобно, например, заполнить на сайте заявку на кредит. Отвечает за функциональность.

User Interface (UI) — это вид интерфейса. Отвечает за статику.

UX-дизайнер — это проектировщик, который изучает потребности пользователей, строит логические схемы работы интерфейса, тестирует прототипы на целевой аудитории и составляет ТЗ для UI-дизайнера.

UI-дизайнер определяет цветовую палитру, располагает объекты в интерфейсе так, чтобы он был удобным и понятным. Визуализирует рабочий прототип, отрисовывает кнопки, формы и другие компоненты. На выходе мы получаем макет.

Специалист по информационной безопасности (Application security)

Анализирует приложение с точки зрения информационной безопасности: определяет требования к приложениям и проверяет реализацию на соответствие этим требованиям. Занимается поиском и анализом уязвимостей.

Разработчик (Developer)

Если говорить о веб-приложении, frontend-разработчик занимается клиентской стороной интерфейса, отвечает за вывод информации для пользователя. Backend-разработчик занят серверной разработкой — тем, чего не видит пользователь, реализацией бизнес-логики приложения. Fullstack — человек, который объединяет в себе обязанности front и back (они тесно связаны).

ПРИМЕР

Вы выбираете рейс из Нью-Йорка в Гонконг. Вы находитесь в зоне frontend. Нажмите клавишу поиска, и вы переместитесь в зону backend, чтобы вам могли правильно вернуть лучший, самый короткий и дешевый рейс в мгновение ока. Как только отобразятся результаты, вы снова окажетесь в зоне frontend.

Отдельно выделены **gamedev-разработчики** (разработчики игр) и **iOS/Android-разработчики** — группы разработчиков под эти ОС.

1C/SAP-разработчики и так далее — разработчики на определенном корпоративном решении. Такие решения покупаются фирмой и дорабатываются под ее нужды, здесь имеются специализированный язык и свои инструменты.

Тестировщик (QA Engineer)

Есть несколько видов тестирования. Соответственно, разные его виды выполняют разные тестировщики.

- По отношению к объекту тестирования.
 1. Функциональное. Тестирование конкретной функциональности, которая была разработана в соответствии с требованиями.
 2. Безопасности. Проверка разработанного продукта на соответствие требованиям безопасности, имитация атаки или несанкционированных действий.
 3. Производительности. Стресс-тестирование, тестирование нагрузки, стабильности.

- По доступу к коду.
 1. Тестирование черного ящика в том случае, когда мы не знаем алгоритмы.
 2. Тестирование белого ящика в том случае, когда учитываются механизмы системы внутри.
 3. Тестирование серого ящика — комбинация первого и второго.

- По степени автоматизации.
 1. Ручное — тест-кейсы.
 2. Автотестирование — специально написанный код для автоматизации проверки определенного кейса.

- По степени изолированности компонента.
 1. Модульное — тестирование определенных модулей системы (компонентов).
 2. Интеграционное — тестирование взаимодействия модулей.
 3. Тестирование системы в целом (полной системы).

Техническая поддержка (Tech Support)

Техническую поддержку часто подразделяют на уровни, чтобы улучшить обслуживание организации или базы клиентов.

Level 1/2/3 support — несколько уровней поддержки приложения по степени знания системы.

На этом этапе обеспечивается общая техническая поддержка, в рамках которой обрабатываются типовые обращения. Цель создания такой службы — обеспечение

постоянной поддержки пользователей на определенном уровне. Значительное число входящих вопросов решается на первой линии.

Level 1 support принимает входящие сообщения заказчиков, отвечает на звонки и письма клиента. Идентифицирует клиента, локализует проблему, обычно следует сценарию. Если может, помогает клиенту на своем уровне. Если отсутствуют даже обходные пути, вопрос передается на следующий уровень поддержки.

Level 2 support — это технические специалисты с более продвинутыми навыками общего или специализированного назначения. Тут происходит систематизация, анализ и решение проблем.

Обязанности специалистов второго уровня:

- контакт с персоналом первой линии и помощь ему;
- фиксация и последующий анализ инцидента;
- решение проблемы;
- передача данных по решенной проблеме на первый уровень.

Если проблема, делегированная специалистам второй линии поддержки, уникальна и квалификация персонала не позволяет ее решить, то в установленные регламентом сроки она должна быть передана дальше — на третий уровень.

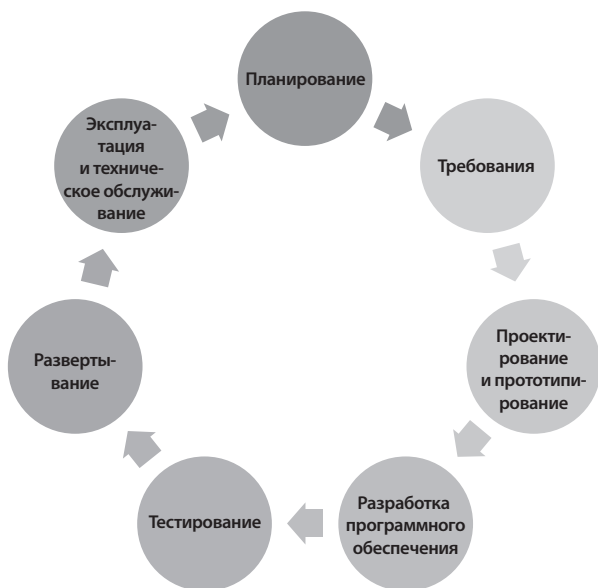
Level 3 support — это обычно команда разработки, которая лучше всех знает продукт и может помочь в решении проблемы.

Жизненный цикл ПО

Жизненный цикл ПО (SDLC, Software Development Life Cycle) — период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.

SDLC — это основные этапы, через которые проходит любой программный продукт. Плюс-минус всегда имеют место следующие этапы.

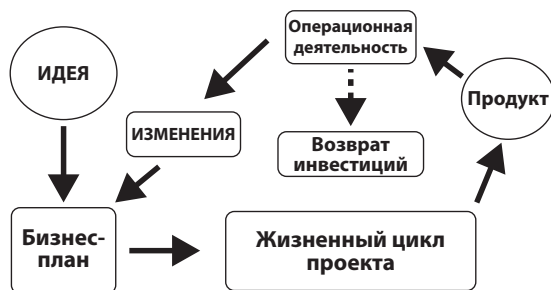
1. Фаза планирования.
2. Фаза сбора требований/анализа.



3. Фаза дизайна/проектирования.
4. Фаза разработки.
5. Фаза тестирования.
6. Фаза внедрения.
7. Фаза поддержки.

ЖЦ продукта начинается с идеи. После того как мы решили создавать новый продукт, мы приступаем к фазе планирования: прорабатываем идею, пишем бизнес-план, проводим анализ конкурентов, рассчитываем предполагаемую прибыль. Далее запускается сам проект по реализации продукта, если руководство решит, что оно того стоит.

После завершения проекта получается продукт, и начинается операционная деятельность по работе с этим продуктом. И далее, если продукт требует изменений, то строится новый бизнес-план и запускается новый проект.



В ЖЦ проекта начальная фаза заключается в переходе идеи в проект, а финальная фаза наступает после подписания актов приема-передачи, когда продукт передается в операционную деятельность. Поэтому можно сказать, что ЖЦ проекта лежит внутри ЖЦ продукта. Это не какие-то строгие правила, стадия планирования и ее содержимое могут быть пропущены: например, если продукт просто требует доработок или новой функциональности.

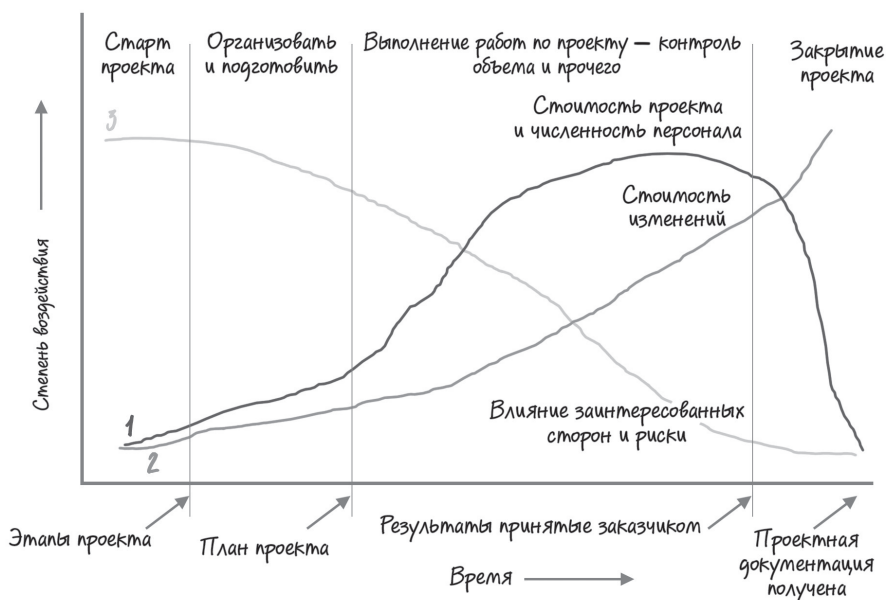
Поэтому хотелось бы отметить, что для продукта и проекта этапы ЖЦ могут отличаться. Это стоит понимать и учитывать.

Жизненный цикл проекта

Любой ЖЦ проекта включает в себя четыре фазы.

1. Начальная — фаза перехода идеи в проект. На этой стадии обычно пишется устав проекта, определяются первичные критерии успеха проекта. Также на этой фазе формируются ключевые роли в проекте: менеджер проекта, владелец, спонсор.

2. Подготовка и организация. На этом этапе выделяются ресурсы для проекта, создается команда, составляется план проекта, определяется бюджет, а также ищутся поставщики услуг, если они необходимы.
3. Выполнение работ. Это фаза создания проекта. Именно на этом этапе создается продукт. Менеджер проекта получает отчеты о ходе выполнения проекта и контролирует процесс. Далее происходит финальное одобрение продукта перед подписанием акта приема работ.
4. Финальная стадия. Основная суть этой фазы — подвести итоги после завершения проекта, подписать акт приема-передачи и передать продукт в операционную деятельность.



На графике показано, как в зависимости от стадии проекта меняются такие показатели, как:

- стоимость проекта и численность персоналом — вовлеченность ресурсов (кривая 1);
- влияние заинтересованных сторон, неопределенность проекта, возможность рисков (кривая 2);
- стоимость изменений по проекту (кривая 3).

1. Мы видим, что самая низкая стоимость изменений — в начале, на этом этапе легче всего повлиять на изменение проекта. Именно здесь определяются цели проекта. Под эти цели выбирается оптимальное решение. Если цели изменятся, то может оказаться, что выбранное решение им не соответствует, и у заказчиков начнутся проблемы.