12+

Кузнецова Юлия Сергеевна

240 ВОПРОСОВ АНАЛИТИКА ТОМ VI

Управление проектами
и продуктами

Юлия Кузнецова

240 вопросов аналитика ТОМ VI. Управление проектами и продуктами

https://litres.ru/74985829
ISBN 9785007148979

Аннотация

Как понять, что продукт действительно нужен клиенту, а проект приносит бизнесу результат? «240 вопросов аналитика» показывает, как мыслить не задачами и функциями, а ценностью и эффектом. Вы узнаете, как проверять гипотезы с помощью MVP, расставлять приоритеты в Product Backlog, оценивать результат продукта и видеть путь клиента через CJM. Книга помогает принимать решения на основе данных и создавать продукты, которые действительно решают проблемы пользователей и бизнеса.

Содержание

Глава 6. Управление проектами и продуктами 6

Вопрос №117. Чем отличается проектное мышление от продуктового? 6

Вопрос №118. Какую роль бизнес-аналитик играет на различных этапах жизненного цикла проекта? 12

Вопрос №119. Чем отличаются роли бизнес-аналитика, системного аналитика, Product Owner и Project Manager? 20

Вопрос №120. Что такое жизненный цикл продукта (Product Lifecycle) и какую роль в нем играет бизнес-аналитик? 29

Вопрос №121. Что такое MVP и как бизнес-аналитику определить минимально жизнеспособный продукт? 37

Вопрос №122. Как бизнес-аналитику работать с Product Backlog и управлять приоритетами продукта? 44

Вопрос №123. Как бизнес-аналитику определять и измерять ценность продукта? 52

Вопрос №124. Что такое Customer Journey Map (CJM) и как бизнес-аналитик использует карту пути клиента? 60

Вопрос №125. Что такое User Story и как бизнес-аналитик правильно их формулирует? 69
Вопрос №126. Чем отличаются User Story, Use Case и функциональные требования? 77
Вопрос №127. Как бизнес-аналитику управлять изменениями требований в проекте? 86
Конец ознакомительного фрагмента. 92

240 вопросов аналитика ТОМ VI Управление проектами и продуктами

Юлия Сергеевна Кузнецова

© Юлия Сергеевна Кузнецова, 2026

ISBN 978-5-0071-4897-9 (т. 6)
ISBN 978-5-0071-3517-7
Создано в интеллектуальной издательской системе Ridero

Глава 6. Управление проектами и продуктами

Вопрос №117. Чем отличается проектное мышление от продуктового?

Почему этот вопрос важен

Сегодня бизнес-аналитик может работать как в проектной команде, так и в продуктовой.

На первый взгляд задачи похожи:

— собирать требования;

— анализировать процессы;

— взаимодействовать с заказчиками;

— помогать разработчикам.

Но подход к работе отличается кардинально.

Аналитик, который не понимает разницы между проектным и продуктовым мышлением, часто предлагает решения, не соответствующие ожиданиям бизнеса.

Главная идея вопроса

Проект ориентирован на создание результата в ограниченный срок, а продукт — на создание долгосрочной ценно-

сти для пользователя и бизнеса.

Что такое проект

Проект — это временная деятельность, направленная на создание уникального результата.

У любого проекта есть:

— начало;

— окончание;

— бюджет;

— сроки;

— конкретный результат.

Примеры проектов:

— внедрение CRM;

— запуск интернет-магазина;

— переход на новую ERP-систему;

— автоматизация склада.

После достижения цели проект завершается.

Что такое продукт

Продукт — это решение, которое продолжает развиваться после первого выпуска. Его задача — постоянно создавать ценность.

Примеры:

— мобильный банк;

— маркетплейс;

— CRM-система как сервис;

— корпоративная платформа;

— мобильное приложение.

После релиза работа только начинается.

Как меняется роль аналитика

В проекте аналитик отвечает за успешную реализацию заранее определенного объема работ.

В продукте аналитик отвечает за постоянное развитие решения.

Он непрерывно изучает:

— пользователей;

— рынок;

— конкурентов;

— поведение клиентов;

— эффективность функций.

Основные различия

Проект.

Главный вопрос: «Как успешно завершить проект?»

Ключевые показатели:

— сроки;

— бюджет;

— объем работ;

— качество реализации.

Продукт.

Главный вопрос: «Как сделать продукт полезнее и успешнее?»

Ключевые показатели:

— удовлетворенность пользователей;

— удержание клиентов;

— рост аудитории;
— прибыль;
— жизненная ценность клиента;
— скорость развития продукта.

Как думает проектный аналитик

Он спрашивает:
— Какие требования нужно реализовать?
— Укладываемся ли мы в сроки?
— Все ли согласовано?
— Выполнен ли объем проекта?

Как думает продуктовый аналитик

Он спрашивает:
— Какую проблему пользователя мы решаем?
— Пользуются ли новой функцией?
— Улучшился ли пользовательский опыт?
— Какие изменения принесут наибольшую ценность?
— Что нужно изменить в следующей версии?

Практический пример.

Компания разрабатывает мобильное приложение банка.

Проектный подход.

Цель: выпустить приложение до установленной даты. После релиза проект считается завершенным.

Продуктовый подход.

После релиза начинается анализ:
— какие функции используют чаще всего;
— где пользователи испытывают сложности;

— почему часть клиентов удаляет приложение;

— какие возможности необходимо добавить.

Работа становится непрерывным циклом улучшений.

Ошибки начинающих аналитиков

Ошибка №1. Считать выпуск продукта завершением работы.

Ошибка №2. Ориентироваться только на выполнение требований.

Ошибка №3. Не анализировать поведение пользователей после внедрения.

Ошибка №4. Не учитывать долгосрочную ценность решений.

Как думает сильный аналитик

Он понимает, что даже идеально реализованная функция бесполезна, если она не приносит ценности пользователям и бизнесу.

Поэтому он оценивает не только успешность реализации, но и успешность использования продукта.

Чек-лист аналитика

☐ Понимать различие между проектом и продуктом.

☐ Знать цели каждого подхода.

☐ Оценивать бизнес-ценность.

☐ Анализировать поведение пользователей.

☐ Измерять результат после внедрения.

☐ Думать о долгосрочном развитии.

Главное правило

Проект заканчивается после достижения поставленной цели. Продукт никогда не заканчивается — он постоянно развивается вместе с потребностями пользователей и изменениями бизнеса. Именно поэтому современный бизнес-аналитик должен одинаково хорошо понимать и проектное, и продуктовое мышление.

Вопрос №118. Какую роль бизнес-аналитик играет на различных этапах жизненного цикла проекта?

Почему этот вопрос важен

Многие начинающие специалисты считают, что работа бизнес-аналитика ограничивается сбором требований.

На практике это лишь небольшая часть его обязанностей.

Современный бизнес-аналитик сопровождает проект практически на всем его жизненном цикле. Он помогает сформулировать бизнес-потребность, определить требования, поддерживает команду во время разработки, участвует в тестировании, внедрении и анализирует результаты после запуска.

Если аналитик включается только на одном этапе, компания рискует потерять целостное понимание задачи, а проект — отклониться от первоначальной бизнес-цели.

Главная идея вопроса

Бизнес-аналитик сопровождает проект от появления идеи до оценки полученного результата, обеспечивая связь между бизнес-целями, требованиями, разработкой и конечной ценностью для компании.

Что такое жизненный цикл проекта

Любой проект проходит несколько последовательных эта-

пов.

Возникновение идеи
↓
Анализ проблемы
↓
Определение требований
↓
Проектирование решения
↓
Разработка
↓
Тестирование
↓
Внедрение
↓
Эксплуатация
↓
Оценка результатов

На каждом этапе аналитик решает разные задачи.

Этап 1. Инициация проекта.

Все начинается не с разработки, а с понимания проблемы.

На этом этапе аналитик помогает ответить на вопросы:

— Какую проблему необходимо решить?

— Почему возникла эта проблема?

— Какие цели стоят перед бизнесом?

— Какой результат считается успешным?

Иногда уже на этом этапе становится понятно, что про-

ект вообще не нужен, а проблему можно решить изменением процесса или организационных правил.

Этап 2. Исследование текущего состояния.

После определения проблемы аналитик изучает существующую ситуацию.

Он анализирует:

— бизнес-процессы;

— информационные системы;

— документы;

— участников процесса;

— ограничения;

— показатели эффективности.

Цель — понять, как работает система сегодня и какие причины вызывают проблему.

Этап 3. Сбор и анализ требований.

Это один из самых известных этапов работы аналитика.

Используются различные методы:

— интервью;

— воркшопы;

— наблюдение;

— анализ документов;

— моделирование процессов;

— изучение данных.

Задача аналитика — превратить пожелания заинтересованных сторон в четкие, непротиворечивые и проверяемые требования.

Этап 4. Проектирование будущего решения.

После анализа требований необходимо определить, каким будет новое решение.

Аналитик:

— моделирует будущие процессы;

— описывает пользовательские сценарии;

— определяет бизнес-правила;

— согласовывает изменения;

— оценивает влияние на другие процессы.

На этом этапе важно убедиться, что предлагаемое решение действительно устраняет первопричину проблемы.

Этап 5. Поддержка разработки.

После передачи требований работа аналитика не заканчивается.

Во время разработки возникают вопросы:

— как трактовать отдельные требования;

— как должна работать бизнес-логика;

— какие сценарии являются исключениями.

Аналитик помогает разработчикам правильно интерпретировать требования и предотвращает ошибки, вызванные неправильным пониманием задачи.

Этап 6. Участие в тестировании.

После завершения разработки аналитик проверяет, соответствует ли реализованное решение первоначальным требованиям.

Он участвует в:

— подготовке сценариев тестирования;
— проверке бизнес-логики;
— приемочном тестировании;
— анализе найденных дефектов.

Важно понимать, что технически работающая функция не всегда решает бизнес-задачу.

Этап 7. Поддержка внедрения.

Во время запуска системы аналитик помогает:
— обучать пользователей;
— отвечать на вопросы;
— уточнять процессы;
— сопровождать первые этапы эксплуатации.

На этом этапе особенно важно получать обратную связь от пользователей.

Этап 8. Оценка результатов.

После внедрения необходимо понять, достиг ли проект поставленных целей.

Для этого аналитик сравнивает:
— плановые показатели;
— фактические результаты;
— экономический эффект;
— удовлетворенность пользователей;
— влияние изменений на бизнес.

Если ожидаемый эффект не достигнут, аналитик помогает определить причины и предложить корректирующие действия.

Практический пример №1

Компания решила внедрить электронный документооборот.

Аналитик:

— изучил существующие процессы;

— выявил узкие места;

— подготовил требования;

— сопровождал разработку;

— участвовал в тестировании;

— помогал при запуске системы;

— оценил сокращение времени согласования документов после внедрения.

Именно комплексное участие позволило добиться ожидаемого результата.

Практический пример №2

Во время разработки CRM-системы заказчик попросил изменить часть требований. Если бы разработчики внесли изменения самостоятельно, возникли бы противоречия с уже реализованными функциями. Я оценила влияние изменений на весь проект, согласовала новые требования и обновила документацию. Благодаря этому удалось избежать дорогостоящих доработок после запуска.

Практический пример №3

После внедрения нового сервиса выяснилось, что сотрудники продолжают использовать старый процесс. Технически система работала без ошибок. Проблема оказалась органи-

зационной: пользователи не понимали преимуществ нового решения. Инициировала дополнительное обучение, переработала инструкции и собрала предложения пользователей. Через несколько недель уровень использования новой системы значительно вырос.

Типичные ошибки аналитиков

Ошибка №1. Считать передачу требований завершением своей работы.

Ошибка №2. Не участвовать в обсуждениях во время разработки.

Ошибка №3. Не проверять соответствие результата бизнес-целям.

Ошибка №4. Игнорировать обратную связь пользователей после внедрения.

Ошибка №5. Не анализировать достигнутый эффект после завершения проекта.

Как думает сильный аналитик

Он рассматривает проект как единый процесс. Для него важно не только правильно описать требования, но и убедиться, что итоговое решение действительно помогло бизнесу. Поэтому он сопровождает проект на всех ключевых этапах и постоянно задает вопрос: «Приближает ли текущее решение нас к той бизнес-цели, ради которой начинался проект?»

Чек-лист аналитика

☐ Понята бизнес-проблема.
☐ Изучено текущее состояние.
☐ Собраны и согласованы требования.
☐ Смоделировано будущее решение.
☐ Поддерживается команда разработки.
☐ Проверена бизнес-логика при тестировании.
☐ Организована поддержка внедрения.
☐ Измерен фактический бизнес-результат.

Главное правило

Работа бизнес-аналитика не начинается со сбором требований и не заканчивается их передачей разработчикам. Его главная ответственность — сопровождать проект на всем жизненном цикле и обеспечивать достижение бизнес-результата, ради которого этот проект был создан.

Вопрос №119. Чем отличаются роли бизнес-аналитика, системного аналитика, Product Owner и Project Manager?

Почему этот вопрос важен

Одна из самых распространенных проблем в проектах — непонимание ролей участников команды.

Очень часто можно услышать:

— «Это должен сделать аналитик».

— «Нет, это задача Product Owner».

— «Этим занимается руководитель проекта».

— «Это ответственность системного аналитика».

Из-за такого смешения обязанностей возникают:

— дублирование работы;

— конфликты;

— потеря времени;

— ошибки в требованиях;

— снижение эффективности команды.

Сильный бизнес-аналитик должен понимать не только свою роль, но и то, как она связана с другими участниками проекта.

Главная идея вопроса

Каждая роль в проекте отвечает за свой уровень принятия

решений. Бизнес-аналитик определяет, что необходимо бизнесу. Системный аналитик определяет, как это реализовать. Product Owner определяет, что приносит максимальную ценность продукту. Project Manager обеспечивает успешное выполнение проекта.

Почему возникает путаница

Во многих компаниях обязанности распределяются поразному.

В небольших командах один человек может совмещать несколько ролей.

Например:

— бизнес-аналитик одновременно выполняет функции Product Owner;

— системный аналитик участвует в управлении требованиями;

— Project Manager помогает приоритизировать задачи.

Поэтому важно понимать не название должности, а ответственность каждой роли.

Роль бизнес-аналитика

Главная задача: понять потребности бизнеса и превратить их в понятные требования.

Бизнес-аналитик отвечает на вопросы:

— Какую проблему необходимо решить?

— Почему она возникла?

— Какую ценность принесет решение?

— Какие процессы необходимо изменить?

— Какие требования существуют?

Основные обязанности:

— анализ бизнеса;

— интервью с заинтересованными сторонами;

— моделирование процессов;

— анализ требований;

— оценка бизнес-ценности;

— сопровождение изменений.

Главный фокус: бизнес и его цели.

Роль системного аналитика

Главная задача: превратить бизнес-требования в техническое решение.

Системный аналитик отвечает на вопросы:

— Как должна работать система?

— Какие данные необходимо хранить?

— Как будут взаимодействовать сервисы?

— Какие API нужны?

— Какие ограничения существуют?

Основные обязанности:

— проектирование решений;

— описание интеграций;

— моделирование данных;

— описание интерфейсов;

— взаимодействие с разработчиками.

Главный фокус: информационная система и ее архитектура.

Роль Product Owner

Главная задача: максимизировать ценность продукта.

Product Owner принимает решения:

— что разрабатывать;

— в каком порядке;

— какие функции важнее;

— какие задачи попадут в следующий релиз.

Он отвечает за:

— видение продукта;

— приоритизацию backlog;

— взаимодействие с клиентами;

— оценку бизнес-ценности функций.

Главный фокус: развитие продукта.

Роль Project Manager

Главная задача: успешно реализовать проект.

Project Manager отвечает:

— за сроки;

— бюджет;

— ресурсы;

— коммуникации;

— управление рисками;

— выполнение плана проекта.

Он обеспечивает условия, при которых команда сможет успешно выполнить работу.

Главный фокус: управление проектом.

Как взаимодействуют эти роли

Представим, что компания хочет внедрить новую систему обработки заказов.

Бизнес-аналитик.

Выясняет:

— какие проблемы есть в текущем процессе;

— какие цели ставит руководство;

— какие требования предъявляют пользователи.

Product Owner.

Определяет:

— какие функции принесут максимальную пользу;

— что войдет в ближайший релиз;

— какие задачи имеют наивысший приоритет.

Системный аналитик.

Проектирует:

— структуру данных;

— взаимодействие сервисов;

— техническую архитектуру;

— интеграции.

Project Manager.

Организует:

— работу команды;

— сроки;

— ресурсы;

— контроль выполнения проекта.

Простая аналогия

Представьте строительство нового офиса.

Бизнес-аналитик выясняет, что именно нужно компании:

— сколько сотрудников будет работать;

— какие помещения необходимы;

— какие бизнес-задачи должен решать новый офис.

Product Owner определяет, какие функции наиболее важны для будущих пользователей офиса и что необходимо реализовать в первую очередь.

Системный аналитик разрабатывает технические решения:

— инженерные сети;

— системы безопасности;

— коммуникации;

— внутреннюю инфраструктуру.

Project Manager следит за тем, чтобы строительство завершилось вовремя, в рамках бюджета и с необходимым качеством.

Где роли пересекаются

Несмотря на различия, участники постоянно взаимодействуют.

Например:

Бизнес-аналитик предлагает несколько вариантов решения.

Product Owner выбирает наиболее ценный вариант.

Системный аналитик оценивает техническую реализуемость.

Project Manager оценивает влияние решения на сроки,

стоимость и ресурсы.

Именно совместная работа позволяет принимать взвешенные решения.

Практический пример

Компания решила добавить функцию онлайн-подписания договоров.

Бизнес-аналитик определил, что главная проблема — длительное согласование документов.

Product Owner решил, что эта функция станет приоритетом следующего релиза.

Системный аналитик разработал схему интеграции с сервисом электронной подписи и подготовил технические требования.

Project Manager включил работы в план проекта, распределил ресурсы и организовал контроль сроков.

Каждый выполнил свою часть работы, и проект был успешно завершен.

Типичные ошибки

Ошибка №1. Бизнес-аналитик начинает самостоятельно принимать продуктовые решения.

Ошибка №2. Системный аналитик определяет бизнес-приоритеты без участия бизнеса.

Ошибка №3. Product Owner подробно проектирует техническую архитектуру.

Ошибка №4. Project Manager самостоятельно изменяет бизнес-требования ради соблюдения сроков.

Ошибка №5. Команда не понимает зоны ответственности друг друга.

Как думает сильный аналитик

Сильный аналитик понимает, что успех проекта зависит не только от качества его собственной работы.

Он знает:

— где заканчивается его ответственность;

— когда необходимо привлечь других специалистов;

— как объединить экспертизу разных ролей для достижения общей цели.

Он не конкурирует с другими участниками проекта, а помогает им работать как единая команда.

Чек-лист аналитика

☐ Понимать ответственность каждой роли.

☐ Не смешивать бизнес- и технические задачи.

☐ Работать совместно с Product Owner.

☐ Поддерживать взаимодействие с системным аналитиком.

☐ Помогать Project Manager управлять изменениями.

☐ Помнить, что общая цель важнее границ отдельных ролей.

Главное правило

Сильная проектная команда строится не на универсальных специалистах, а на четком понимании ролей и эффек-

тивном взаимодействии между ними. Бизнес-аналитик отвечает за понимание потребностей бизнеса, системный аналитик — за техническое решение, Product Owner — за ценность продукта, а Project Manager — за успешную реализацию проекта. Именно совместная работа этих ролей позволяет создавать решения, которые приносят реальную пользу бизнесу.

Вопрос №120. Что такое жизненный цикл продукта (Product Lifecycle) и какую роль в нем играет бизнес-аналитик?

Почему этот вопрос важен

Многие начинающие аналитики сосредотачиваются исключительно на проектах. Они думают: «Мы разработали систему, внедрили ее — работа закончена». Но в современном бизнесе большинство цифровых решений являются продуктами. После первого релиза начинается самая длинная и самая важная часть их жизни — развитие.

Мобильное приложение банка, интернет-магазин, CRM-система, маркетплейс или корпоративный портал постоянно меняются. Появляются новые функции, меняются потребности клиентов, усиливается конкуренция, развиваются технологии.

Именно поэтому бизнес-аналитик должен понимать не только жизненный цикл проекта, но и жизненный цикл продукта.

Главная идея вопроса

Жизненный цикл продукта — это непрерывный процесс создания, развития, совершенствования и, при необходимости, завершения продукта. На каждом этапе бизнес-анали-

тик помогает принимать решения, основанные на данных, потребностях пользователей и целях бизнеса.

Что такое жизненный цикл продукта

Любой продукт проходит несколько этапов.

Идея
↓
Исследование
↓
Создание
↓
Запуск
↓
Рост
↓
Зрелость
↓
Трансформация или завершение

Каждый этап требует разных управленческих решений.

Этап 1. Возникновение идеи.

Любой продукт начинается с проблемы. Не с технологии. Не с красивой идеи. А с вопроса: Какую проблему пользователя мы хотим решить?

На этом этапе аналитик:

— изучает рынок;

— анализирует конкурентов;

— исследует потребности клиентов;

— оценивает потенциальную ценность продукта;

— помогает сформулировать бизнес-гипотезу.

Главная цель — понять, стоит ли вообще создавать продукт.

Этап 2. Исследование.

После появления идеи необходимо проверить ее жизнеспособность.

Аналитик помогает ответить на вопросы:

— существует ли реальный спрос;

— кто будет пользоваться продуктом;

— какие альтернативы уже есть на рынке;

— какие риски могут возникнуть.

Инструменты:

— интервью;

— опросы;

— анализ рынка;

— Customer Journey Map;

— конкурентный анализ;

— исследование данных.

Этап 3. Создание продукта.

На этом этапе начинается разработка.

Бизнес-аналитик:

— собирает требования;

— моделирует процессы;

— описывает пользовательские сценарии;

— помогает Product Owner определить минимально жизнеспособный продукт (MVP);

— согласовывает требования с заинтересованными сторонами.

Главная задача — создать продукт, который действительно решает проблему пользователя.

Этап 4. Запуск.

После разработки продукт выходит к пользователям. Теперь важно не только то, что было реализовано, но и то, как продукт используется.

Аналитик анализирует:

— количество пользователей;

— первые отзывы;

— возникающие проблемы;

— показатели использования функций;

— технические ошибки.

Очень часто именно первые недели после запуска дают наиболее ценную информацию.

Этап 5. Рост.

Если продукт оказался востребованным, начинается этап развития.

На этом этапе аналитик отвечает на вопросы:

— какие функции наиболее востребованы;

— какие возможности необходимо добавить;

— где пользователи испытывают сложности;

— какие процессы требуют оптимизации.

Работа становится непрерывной. Каждый новый релиз строится на анализе предыдущего опыта.

Этап 6. Зрелость.

Со временем продукт становится стабильным.

Основные задачи меняются.

Теперь аналитик помогает:

— удерживать клиентов;

— повышать эффективность;

— снижать затраты;

— искать новые источники дохода;

— улучшать пользовательский опыт.

На этом этапе большое значение приобретают анализ поведения пользователей и поиск возможностей для постепенного совершенствования продукта.

Этап 7. Трансформация или завершение.

Ни один продукт не существует вечно. Рано или поздно появляются новые технологии, меняются ожидания клиентов и бизнес-модель компании.

Перед организацией возникает выбор:

— модернизировать продукт;

— полностью переработать его;

— заменить новым решением;

— прекратить поддержку.

Бизнес-аналитик оценивает:

— стоимость дальнейшего развития;

— влияние на клиентов;

— риски перехода;

— экономическую целесообразность.

Роль аналитика на протяжении всего жизненного цикла

На каждом этапе его задачи различаются. Во время исследования он ищет проблемы. Во время разработки помогает создать правильное решение. После запуска анализирует результаты. Во время роста ищет новые возможности. На этапе зрелости повышает эффективность. При трансформации оценивает стратегические решения. Таким образом, аналитик сопровождает продукт на протяжении всей его жизни.

Практический пример №1

Компания разработала мобильное приложение для доставки еды.

После запуска заметила, что большинство пользователей прекращают оформление заказа на этапе оплаты. Дополнительное исследование показало, что процесс оплаты слишком сложен. После его упрощения количество завершенных заказов значительно выросло. Если бы команда ограничилась только успешным релизом, эта проблема могла бы остаться незамеченной.

Практический пример №2

Онлайн-банк несколько лет успешно развивал мобильное приложение.

Я регулярно анализировала поведение клиентов и обнаружил рост использования переводов по номеру телефона. Компания сосредоточила развитие именно на этом направлении, что привело к увеличению активности пользователей

и укреплению конкурентных преимуществ.

Практический пример №3

CRM-система использовалась более десяти лет. Поддержка старой архитектуры становилась все дороже. Подготовила сравнительный анализ затрат на модернизацию и разработку новой платформы. На основе этого анализа руководство приняло решение о постепенном переходе на современную систему, снизив риски и сохранив непрерывность работы бизнеса.

Типичные ошибки аналитиков

Ошибка №1. Считать выпуск продукта завершением работы.

Ошибка №2. Не анализировать поведение пользователей после запуска.

Ошибка №3. Разрабатывать новые функции без изучения данных.

Ошибка №4. Игнорировать изменения рынка и конкурентов.

Ошибка №5. Не оценивать жизнеспособность продукта в долгосрочной перспективе.

Как думает сильный аналитик

Он понимает, что настоящий успех продукта определяется не количеством реализованных функций, а тем, насколько хорошо продукт продолжает решать проблемы пользователей через месяцы и годы после запуска.

Поэтому аналитик постоянно наблюдает за продуктом, со-

бирает обратную связь, анализирует данные и помогает бизнесу принимать решения о дальнейшем развитии.

Чек-лист аналитика

☐ Определить проблему пользователя.
☐ Проверить жизнеспособность идеи.
☐ Собрать качественные требования.
☐ Проанализировать результаты запуска.
☐ Изучать поведение пользователей.
☐ Принимать решения на основе данных.
☐ Постоянно совершенствовать продукт.
☐ Планировать долгосрочное развитие.

Главное правило

Продукт не заканчивается первым релизом. Настоящая работа начинается после того, как им начинают пользоваться реальные люди. Именно поэтому бизнес-аналитик сопровождает продукт на протяжении всего жизненного цикла, помогая компании постоянно создавать ценность для пользователей и бизнеса.

Вопрос №121. Что такое MVP и как бизнес-аналитику определить минимально жизнеспособный продукт?

Почему этот вопрос важен

Одна из самых частых ошибок при создании новых продуктов — попытка сразу сделать идеальное решение.

Команды добавляют десятки функций:

— расширенные настройки;

— дополнительные возможности;

— сложные сценарии;

— интеграции;

— улучшения интерфейса.

В результате разработка занимает месяцы или годы, а бизнес еще не знает, нужен ли этот продукт пользователям. Именно поэтому используется подход MVP.

Главная идея вопроса

MVP (Minimum Viable Product) — это минимальная версия продукта, которая позволяет проверить главную гипотезу и получить обратную связь от реальных пользователей с минимальными затратами ресурсов.

Что такое MVP

MVP — это не плохая или недоработанная версия про-

дукта. Это решение с минимальным набором функций, достаточным для проверки ценности.

Его цель: не создать окончательный продукт, а получить ответы на важные вопросы:

— Нужен ли продукт пользователям?

— Решает ли он реальную проблему?

— Готовы ли люди им пользоваться?

— Какие изменения необходимы дальше?

Чем MVP отличается от обычного продукта

Представим, что компания хочет создать сервис доставки еды.

Полная версия может включать:

— мобильное приложение;

— программу лояльности;

— рекомендации;

— подписки;

— интеграцию с банками;

— систему отзывов;

— персональные предложения.

Но для проверки основной идеи может быть достаточно:

— каталога ресторанов;

— возможности выбрать блюдо;

— оформления заказа;

— доставки.

Это и будет MVP.

Главный принцип MVP

Создать минимальное решение, которое проверяет главную бизнес-гипотезу.

Не: «Какие функции мы можем добавить?»

А: «Какую главную проблему пользователя мы должны проверить?»

Роль бизнес-аналитика в создании MVP

Бизнес-аналитик помогает определить:

— какую проблему необходимо решить;

— кто является пользователем;

— какие функции являются обязательными;

— какие функции можно отложить;

— как измерить успех.

Как определить MVP

Шаг 1. Определить проблему.

Первый вопрос: Почему вообще нужен этот продукт?

Например:

Проблема: клиенты долго оформляют банковские услуги.

Шаг 2. Определить целевого пользователя.

Необходимо понять:

— для кого создается продукт;

— какие потребности есть у пользователя;

— какие сложности он испытывает.

Шаг 3. Определить основную ценность.

Нужно ответить: «Почему пользователь должен выбрать этот продукт?»

Например:

Не: «Создать мобильное приложение банка».

А: «Позволить клиенту открыть счет без посещения офиса».

Шаг 4. Выбрать минимальный набор функций.

Каждая функция должна отвечать на вопрос: Помогает ли она проверить основную гипотезу? Если нет — ее можно отложить.

Шаг 5. Определить критерии успеха.

До запуска необходимо понять: как мы узнаем, что MVP успешен?

Например:

— 10 000 пользователей зарегистрировались;

— 30% пользователей завершили первый заказ;

— время оформления сократилось на 50%.

MVP и приоритизация требований

Бизнес-аналитик часто сталкивается с большим количеством пожеланий.

Например:

Заказчик хочет добавить:

— чат;

— рекомендации;

— рейтинги;

— сложную аналитику;

— интеграции.

Но аналитик должен определить: что необходимо сейчас, а что можно реализовать позже.

Практический пример №1

Онлайн-магазин.

Компания хочет создать новый маркетплейс.

Первоначальная идея: создать полноценную платформу.

Аналитик предложил начать с MVP:

— несколько категорий товаров;

— простой каталог;

— оформление заказа;

— базовая система оплаты.

После запуска стало понятно:

— какие товары наиболее востребованы;

— какие функции нужны пользователям;

— какие направления развития перспективны.

Компания избежала больших затрат на ненужные возможности.

Практический пример №2

Корпоративная система.

Компания планировала сложную систему управления персоналом. Вместо полной автоматизации всех процессов аналитик предложил начать с одного наиболее проблемного процесса — оформления отпусков.

После успешного запуска команда получила:

— обратную связь пользователей;

— данные об эффективности;

— понимание следующих шагов.

Практический пример №3

Мобильное приложение.

Компания хотела создать приложение для клиентов. Первоначальный список требований содержал более 100 функций. Аналитик провел анализ и выяснил: главная проблема клиентов — невозможность быстро получить информацию о заказе.

В MVP вошли только:

— просмотр заказа;

— уведомления;

— история операций.

После подтверждения ценности продукт начали расширять.

Типичные ошибки при создании MVP

Ошибка №1. Путать MVP с плохим качеством.

MVP должен быть минимальным, но полезным.

Ошибка №2. Добавлять слишком много функций.

Это превращает MVP в обычную разработку.

Ошибка №3. Не определять гипотезу.

Без гипотезы невозможно понять результат.

Ошибка №4. Не измерять успех.

Если нет критериев, невозможно сделать выводы.

Ошибка №5. Игнорировать обратную связь пользователей.

Главная цель MVP — обучение.

Как думает сильный аналитик

Он понимает: лучший продукт создается не путем пред-

положений, а через постоянную проверку гипотез.

Поэтому он задает вопросы:

— Какую проблему мы проверяем?

— Какой минимальный результат нам нужен?

— Какие функции действительно необходимы?

— Какие данные покажут успех?

— Что мы будем делать после получения обратной связи?

Чек-лист определения MVP

☐ Определена проблема пользователя.

☐ Понятна целевая аудитория.

☐ Сформулирована основная гипотеза.

☐ Выбраны минимально необходимые функции.

☐ Определены критерии успеха.

☐ Запланирован сбор обратной связи.

☐ Определены следующие шаги развития.

Главное правило

MVP — это не минимальный продукт, а минимальный способ проверить ценность идеи. Сильный бизнес-аналитик помогает команде не тратить ресурсы на создание ненужных функций, а быстро получить знания о пользователях и принять более качественные решения о развитии продукта.

Вопрос №122. Как бизнес-аналитику работать с Product Backlog и управлять приоритетами продукта?

Почему этот вопрос важен

В продуктовой разработке команда почти всегда сталкивается с одной проблемой: идей и задач больше, чем ресурсов для их реализации. Пользователи хотят новые функции. Руководители хотят улучшить бизнес-показатели. Команда разработки предлагает технические улучшения. Все считают свои задачи важными. Но невозможно сделать всё одновременно. Именно поэтому необходимо управлять Product Backlog — списком всех идей, требований, улучшений и задач продукта.

Главная идея вопроса

Product Backlog — это управляемый список задач продукта, а приоритизация помогает определить, какие изменения принесут максимальную ценность бизнесу и пользователям в первую очередь.

Что такое Product Backlog

Product Backlog — это единый список всех возможных изменений продукта.

В него могут входить:

— новые функции;

— улучшение существующих возможностей;
— исправление ошибок;
— технические улучшения;
— требования пользователей;
— идеи развития.

Пример Backlog интернет-магазина:

— добавить оплату через новый сервис;
— улучшить поиск товаров;
— ускорить загрузку страниц;
— изменить личный кабинет;
— исправить ошибку оформления заказа;
— добавить рекомендации товаров.

Но наличие задачи в Backlog не означает, что она будет реализована сразу. Каждая задача должна пройти оценку.

Почему нельзя делать задачи просто по порядку поступления

Представим ситуацию: У компании есть пять задач.

Задача №1. Изменить цвет кнопки в приложении.

Задача №2. Сократить время оформления заказа на 50%.

Задача №3. Добавить новый отчет для одного руководителя.

Задача №4. Исправить ошибку, из-за которой теряются заказы.

Задача №5. Добавить новую функцию, которая может увеличить продажи.

Все задачи требуют ресурсов. Но их ценность для бизне-

са совершенно разная. Поэтому нужен механизм приоритизации.

Роль бизнес-аналитика в работе с Backlog

Бизнес-аналитик помогает:

— понять смысл каждой задачи;

— определить бизнес-ценность;

— выявить зависимости;

— оценить влияние изменений;

— подготовить требования;

— предоставить данные для принятия решения.

При этом окончательное решение о приоритетах часто принимает Product Owner.

Основные критерии приоритизации

1. ***Бизнес-ценность.***

Главный вопрос: «Какую пользу принесет эта задача?»

Например:

— увеличит ли она прибыль;

— сократит ли расходы;

— улучшит ли опыт клиента;

— снизит ли риски.

2. ***Влияние на пользователей.***

Некоторые изменения затрагивают тысячи пользователей. Другие — только небольшую группу. Чем больше положительное влияние, тем выше потенциальный приоритет.

3. ***Срочность.***

Некоторые задачи нельзя откладывать.

Например:

— изменение законодательства;

— критическая ошибка;

— нарушение безопасности.

4. ***Стоимость реализации.***

Даже очень ценная идея может быть не первой в очереди, если ее реализация слишком сложная.

Важно сравнивать: ценность результата и затраты на получение результата.

5. ***Риски.***

Иногда задача важна не потому, что приносит прибыль, а потому что снижает угрозы.

Например:

— обновление системы безопасности;

— устранение технического долга;

— переход на новую платформу.

Популярные методы приоритизации

Метод MoSCoW.

Один из самых распространенных методов.

Требования делятся на четыре категории:

Must Have.

Обязательно необходимо.

Без этого продукт не выполняет свою основную задачу.

Should Have.

Очень желательно.

Можно реализовать после обязательных функций.

Could Have.

Можно сделать при наличии ресурсов.

Won't Have.

Не входит в текущий объем работы.

Метод RICE.

Использует четыре параметра:

— Reach — количество затронутых пользователей;

— Impact — влияние;

— Confidence — уверенность в оценке;

— Effort — необходимые усилия.

Позволяет сравнивать разные идеи между собой.

Метод Value vs Effort.

Задачи оцениваются по двум параметрам:

— ценность;

— сложность реализации.

Наиболее привлекательны задачи:

высокая ценность + низкая сложность.

Именно они часто становятся первыми кандидатами на реализацию.

Как аналитик помогает улучшить Backlog

Хороший аналитик следит, чтобы задачи были:

— понятными;

— связанными с бизнес-целями;

— измеримыми;

— достаточно детализированными;

— актуальными.

Плохой элемент Backlog: «Сделать удобнее оформление заказа».

Хороший элемент: «Сократить количество шагов оформления заказа с пяти до трех, чтобы увеличить процент завершенных покупок на 10%».

Практический пример №1

Компания имеет 50 идей для развития мобильного приложения. Руководство хочет реализовать всё.

Аналитик провел оценку:

— влияние на пользователей;

— ожидаемый доход;

— стоимость разработки.

Выяснилось, что пять функций дают 80% потенциального эффекта. Команда сосредоточилась на них.

Практический пример №2

Банк планировал добавить новые возможности в приложение. Однако анализ показал, что большинство пользователей покидают приложение из-за долгой загрузки. Вместо новых функций команда сначала улучшила производительность. В результате выросла удовлетворенность клиентов.

Практический пример №3

Производственная компания накопила большой технический долг. Разработчики хотели полностью переписать систему. Аналитик оценил влияние и показал, какие технические проблемы реально влияют на бизнес. В результате команда исправила наиболее критичные области без остано-

ки работы.

Типичные ошибки аналитиков

Ошибка №1. Считать все требования одинаково важными.

Ошибка №2. Приоритизировать только по мнению руководителя.

Ошибка №4. Не учитывать стоимость реализации.

Ошибка №5. Не пересматривать приоритеты при изменении ситуации.

Как думает сильный аналитик

Он понимает: невозможно реализовать всё.

Поэтому задача аналитика — помочь команде выбрать то, что создаст максимальную ценность.

Он постоянно задает вопросы:

— Какая проблема решается?

— Для кого это важно?

— Какой будет эффект?

— Что произойдет, если отложить эту задачу?

— Есть ли более простой способ получить тот же результат?

Чек-лист работы с Product Backlog

☐ Все задачи связаны с бизнес-целями.

☐ Понятна ценность каждой задачи.

☐ Определены заинтересованные пользователи.

☐ Оценены риски.

☐ Учтена стоимость реализации.
☐ Приоритеты регулярно пересматриваются.
☐ Команда понимает причины выбранных решений.

Главное правило

Product Backlog — это не список желаний, а инструмент управления ценностью продукта. Сильный бизнес-аналитик помогает команде не просто выполнять больше задач, а выбирать именно те изменения, которые максимально приближают бизнес к его целям.

Вопрос №123. Как бизнес-аналитику определять и измерять ценность продукта?

Почему этот вопрос важен

Одна из главных ошибок в разработке продуктов — считать успехом сам факт создания новой функции. Команда может потратить месяцы на разработку. Функция может выглядеть современной. Руководство может быть довольно результатом. Но главный вопрос остается: создала ли эта функция реальную ценность для бизнеса и пользователей? Именно поэтому современный бизнес-аналитик должен уметь не только анализировать требования, но и понимать ценность продукта.

Главная идея вопроса

Ценность продукта — это измеримый результат, который получает бизнес и пользователь благодаря использованию продукта. Аналитик помогает определить эту ценность, сформулировать показатели успеха и проверить, достигнуты ли ожидаемые результаты.

Что такое ценность продукта

Ценность может проявляться по-разному.

Для пользователя:

— решение проблемы;

— экономия времени;
— удобство;
— улучшение опыта;
— получение новых возможностей.

Для бизнеса:
— рост доходов;
— снижение затрат;
— увеличение количества клиентов;
— повышение эффективности;
— снижение рисков.

Почему функции не равны ценности

Рассмотрим пример.

Компания добавила в мобильное приложение новую функцию. Команда говорит: «Функция успешно разработана». Но это только технический результат.

Настоящие вопросы:
— Пользуются ли ей клиенты?
— Стало ли клиентам удобнее?
— Увеличилась ли прибыль?
— Снизилось ли количество обращений в поддержку?

Если ответ отрицательный, функция может не создавать никакой ценности.

Модель создания ценности

Хороший аналитический подход выглядит так:

Проблема пользователя
↓

Решение
↓
Использование продукта
↓
Изменение поведения
↓
Бизнес-результат

Например:

Проблема: Клиенты долго оформляют заказ.

Решение: Упростить процесс покупки.

Изменение поведения: Больше клиентов завершают оформление.

Бизнес-результат: Рост продаж.

Как определить ценность продукта

1. ***Понять проблему.***

Первый вопрос: Какую проблему мы решаем? Если проблема не существует или не является значимой, ценность продукта будет низкой.

2. ***Определить пользователя.***

Необходимо понимать:

— кто использует продукт;

— какие у него потребности;

— какие трудности он испытывает;

— какие альтернативы существуют.

3. ***Сформулировать ожидаемый результат.***

Важно перейти от функций к результатам.

Плохая формулировка: «Добавить кнопку быстрого заказа».

Хорошая: «Сократить время оформления заказа и увеличить количество завершенных покупок».

Метрики ценности продукта

Чтобы оценить результат, необходимо использовать показатели.

1. ***Финансовые показатели.***

Например:

— выручка;

— прибыль;

— средний чек;

— стоимость привлечения клиента;

— экономия затрат.

2. ***Пользовательские показатели.***

Например:

— количество активных пользователей;

— удержание клиентов;

— удовлетворенность пользователей;

— количество повторных покупок.

3. ***Поведенческие показатели.***

Они показывают, как люди используют продукт.

Например:

— частота использования функции;

— количество завершенных действий;

— время выполнения операции;

— процент отказов.

4. ***Операционные показатели.***

Показывают влияние продукта на внутренние процессы.

Например:

— скорость обработки заявок;

— количество ошибок;

— нагрузка на сотрудников;

— время выполнения операций.

Роль бизнес-аналитика в измерении ценности

Аналитик помогает:

— определить правильные метрики;

— связать функции с бизнес-целями;

— анализировать данные после внедрения;

— находить причины отклонений;

— предлагать улучшения.

Он должен уметь ответить:

— Почему мы создаем эту функцию?

— Как поймем, что она успешна?

— Какие показатели должны измениться?

— Через какое время оценивать эффект?

Практический пример №1

Интернет-магазин.

Компания решила добавить рекомендации товаров.

Команда ожидала:

— рост продаж;

— увеличение среднего чека.

После запуска аналитик проверил данные.

Оказалось:

— пользователи видят рекомендации;

— но почти не переходят по ним.

Причина: рекомендации были недостаточно персонализированы.

После изменения алгоритма выросло количество покупок через рекомендации. Ценность была создана не самим фактом добавления функции, а достижением бизнес-результата.

Практический пример №2

Банковское приложение.

Банк добавил возможность открытия счета онлайн.

Метрика успеха: не количество созданных кнопок и экранов.

А:

— сколько клиентов открыли счет;

— сколько завершили процесс;

— насколько сократилось время обслуживания.

После анализа выяснилось, что основная проблема была не в отсутствии функции, а в сложной процедуре подтверждения личности. Изменение процесса дало больший эффект, чем добавление новых возможностей.

Практический пример №3

Корпоративная система.

Компания внедрила новую CRM.

Проект считался успешным после запуска системы.

Но аналитик проверил результаты через три месяца.

Выяснилось:

— сотрудники вводят меньше данных, чем ожидалось;

— продажи не выросли;

— часть процессов осталась прежней.

Было принято решение изменить настройки системы и обучить сотрудников. Только после этого инвестиции начали приносить ожидаемый результат.

Типичные ошибки аналитиков

Ошибка №1. Измерять количество реализованных функций.

Ошибка №2. Не связывать требования с бизнес-целями.

Ошибка №3. Оценивать результат сразу после запуска.

Ошибка №4. Игнорировать мнение пользователей.

Ошибка №5. Не анализировать фактические данные.

Как думает сильный аналитик

Он понимает: продукт существует не ради функций. Функции — это инструменты. Главная цель — изменить ситуацию к лучшему.

Поэтому аналитик постоянно задает вопросы:

— Какую пользу получает пользователь?

— Какой эффект получает бизнес?

— Какие данные подтверждают успех?

— Нужно ли изменить решение после получения новых результатов?

Чек-лист оценки ценности продукта

☐ Понятна проблема пользователя.
☐ Определена целевая аудитория.
☐ Сформулирован ожидаемый результат.
☐ Выбраны измеримые метрики.
☐ Проведен анализ после внедрения.
☐ Подтвержден бизнес-эффект.
☐ Определены следующие шаги развития.

Главное правило

Успешный продукт определяется не количеством функций и не фактом его создания. Настоящий успех измеряется тем, какую ценность он создает для пользователей и бизнеса. Задача бизнес-аналитика — постоянно связывать решения с результатами и помогать команде создавать не просто работающие продукты, а действительно полезные продукты.

Вопрос №124. Что такое Customer Journey Map (CJM) и как бизнес-аналитик использует карту пути клиента?

Почему этот вопрос важен

Многие компании создают продукты, ориентируясь только на внутренние процессы.

Они думают:

— как устроена система;

— какие функции можно добавить;

— какие операции выполняют сотрудники.

Но клиент оценивает продукт совершенно иначе.

Его интересует:

— насколько легко решить свою задачу;

— сколько времени это займет;

— какие трудности возникнут;

— насколько удобным будет весь опыт взаимодействия.

Именно поэтому бизнес-аналитику важно понимать путь клиента от первого контакта с компанией до получения результата.

Главная идея вопроса

Customer Journey Map (CJM) — это визуальная карта пути клиента, которая показывает все этапы взаимодействия

пользователя с компанией, его действия, эмоции, проблемы и возможности для улучшения опыта.

Что такое путь клиента

Путь клиента — это последовательность действий, которые человек проходит для достижения своей цели.

Например, клиент хочет купить автомобиль.

Его путь может выглядеть так:

Возникновение потребности
↓
Поиск информации
↓
Сравнение вариантов
↓
Обращение в компанию
↓
Консультация
↓
Покупка
↓
Получение автомобиля
↓
Послепродажное обслуживание

На каждом этапе клиент получает определенный опыт.

Почему компании используют CJM

Без карты пути клиента организация видит только внутреннюю сторону процесса.

Например:

Компания считает: «Мы быстро обрабатываем заявки».

Но клиент может думать: «Мне приходится ждать ответа несколько дней». Причина разницы — компания смотрит изнутри, а клиент — со своей стороны. CJM помогает увидеть реальную картину.

Основные элементы Customer Journey Map

Обычно карта содержит несколько компонентов.

1. ***Персона клиента.***

Сначала необходимо понять: Кто этот человек?

Например:

— возраст;

— потребности;

— цели;

— мотивация;

— проблемы.

Важно создавать карту не для абстрактного пользователя, а для конкретного типа клиента.

2. ***Этапы пути.***

Определяются основные шаги клиента.

Например, для интернет-магазина:

— поиск товара;

— просмотр каталога;

— добавление в корзину;

— оформление заказа;

— доставка;

— повторная покупка.

3. ***Действия клиента.***
Что делает пользователь на каждом этапе?
Например:
— вводит запрос;
— сравнивает товары;
— читает отзывы;
— обращается в поддержку.
4. ***Точки контакта.***
Это места взаимодействия клиента с компанией.
Например:
— сайт;
— мобильное приложение;
— офис;
— колл-центр;
— электронная почта;
— социальные сети.
5. ***Эмоции клиента.***
Важно понимать: что чувствует человек на каждом этапе.
Например:
Положительные эмоции:
— уверенность;
— удобство;
— доверие.
Негативные эмоции:
— раздражение;
— неопределенность;

— разочарование.

6. ***Проблемы и возможности.***

На каждом этапе аналитик ищет:

— где клиент испытывает сложности;

— что можно улучшить;

— какие изменения принесут пользу.

Роль бизнес-аналитика в создании CJM

Аналитик использует карту пути клиента, чтобы:

— лучше понять потребности пользователей;

— выявить проблемы процесса;

— определить требования к продукту;

— найти возможности улучшения;

— связать клиентский опыт с бизнес-целями.

Как создавать Customer Journey Map

Шаг 1. Определить цель анализа.

Нельзя изучать весь путь клиента одновременно без конкретной задачи.

Например:

— почему пользователи не завершают покупку;

— почему клиенты уходят;

— где возникает неудовлетворенность.

Шаг 2. Собрать данные.

Источники информации:

— интервью с клиентами;

— отзывы;

— аналитика поведения;

— обращения в поддержку;
— данные продаж;
— наблюдение за пользователями.

Шаг 3. Построить карту.

Определяются:
— этапы пути;
— действия клиента;
— проблемы;
— эмоции;
— точки улучшения.

Шаг 4. Найти проблемные зоны.

После построения карты аналитик ищет:
— наиболее болезненные этапы;
— причины отказов;
— возможности улучшения.

Шаг 5. Сформировать решения.

Каждая проблема должна привести к действию.

Например:

Проблема: Клиенты бросают оформление заказа.

Решение: Сократить количество шагов покупки.

Практический пример №1

Интернет-магазин.

Компания заметила снижение продаж. Первоначально предполагали: «Нужно больше рекламы». Построила CJM.

Выяснилось:
— пользователи находят товар;

— добавляют его в корзину;

— но уходят на этапе оплаты.

Причина: слишком сложная процедура регистрации. После упрощения оформления количество завершенных заказов выросло.

Практический пример №2

Банк.

Банк изучил путь клиента при оформлении кредита. CJM показала: самая большая проблема возникала не при принятии решения, а во время сбора документов.

Компания изменила процесс:

— добавила автоматическую проверку данных;

— сократила количество документов;

— улучшила информирование клиента.

Время оформления кредита уменьшилось.

Практический пример №3

Медицинская клиника.

Клиника анализировала путь пациента.

Выяснилось: основная проблема была не в качестве лечения, а в ожидании между этапами:

— запись;

— прием;

— получение результатов.

После оптимизации процесса повысилась удовлетворенность пациентов.

Типичные ошибки при использовании CJM

Ошибка №1. Создавать карту без реальных данных.

Ошибка №2. Смотреть только на действия клиента и игнорировать эмоции.

Ошибка №3. Описывать внутренний процесс компании вместо пути пользователя.

Ошибка №4. Создать карту и не использовать ее для изменений.

Ошибка №5. Изучать всех клиентов одинаково.

Как думает сильный аналитик

Он понимает: бизнес существует не только внутри компании. Настоящая ценность создается в момент взаимодействия клиента с продуктом или услугой.

Поэтому аналитик постоянно задает вопросы:

— Что хочет получить клиент?

— Где возникает проблема?

— Почему клиент испытывает трудности?

— Как изменение процесса повлияет на опыт пользователя?

— Как измерить улучшение?

Чек-лист создания Customer Journey Map

☐ Определена персона клиента.

☐ Понятна цель пользователя.

☐ Описаны этапы пути.

☐ Найдены точки контакта.

☐ Собраны реальные данные.

☐ Определены проблемы клиента.
☐ Найдены возможности улучшения.
☐ Измерен эффект изменений.

Главное правило

Customer Journey Map помогает бизнесу посмотреть на свои процессы глазами клиента. Сильный бизнес-аналитик использует CJM не просто как красивую схему, а как инструмент поиска проблем, формирования требований и создания решений, которые делают продукт действительно ценным для пользователя.

Вопрос №125. Что такое User Story и как бизнес-аналитик правильно их формулирует?

Почему этот вопрос важен

В современных продуктовых командах недостаточно просто написать список функций.

Фраза: «Добавить кнопку оплаты»

не объясняет:

— кому нужна эта возможность;

— какую проблему она решает;

— почему она важна;

— какой результат ожидается.

Команда может реализовать функцию технически правильно, но не решить реальную потребность пользователя.

Именно поэтому в Agile-подходах широко используется формат User Story.

Главная идея вопроса

User Story — это краткое описание потребности пользователя, которое показывает, кто хочет получить изменение, что именно ему нужно и какую ценность это принесет.

Что такое User Story

User Story переводится как «пользовательская история».

Это способ описания требований через точку зрения

пользователя.

Классическая формула:

Как [роль пользователя], я хочу [действие], чтобы
[ценность или причина].

Пример простой User Story.

Плохо: Добавить возможность восстановления пароля. Почему плохо?

Не понятно:

— кому это нужно;

— зачем;

— какой результат ожидается.

Хорошо: Как пользователь мобильного приложения, я хочу восстановить пароль самостоятельно, чтобы получить доступ к своему аккаунту без обращения в поддержку.

Здесь понятно:

— кто пользователь;

— что ему нужно;

— зачем это необходимо.

Почему User Story лучше обычного требования

Обычное требование часто описывает только решение.

Например: «Создать форму регистрации из пяти полей». Но пользовательская проблема может быть другой: «Пользователь должен быстро создать аккаунт». Возможно, пять полей вообще не нужны. User Story заставляет команду сначала понять потребность, а уже потом искать решение.

Основные элементы хорошей User Story

1. ***Роль пользователя.***

Необходимо понимать: кто получает ценность.

Примеры:

— клиент;

— оператор поддержки;

— менеджер продаж;

— администратор системы;

— руководитель.

2. ***Действие.***

Что хочет сделать пользователь?

Например:

— оформить заказ;

— получить отчет;

— изменить данные;

— найти информацию.

3. ***Ценность.***

Почему это важно?

Например:

— сэкономить время;

— избежать ошибок;

— быстрее получить результат;

— принять решение.

INVEST — признаки качественной User Story

Хорошая User Story соответствует принципу INVEST.

Independent — независимая.

История должна быть максимально самостоятельной.

Negotiable — обсуждаемая.

Она не является жесткой технической инструкцией. Команда может обсуждать лучший способ реализации.

Valuable — ценная.

Она должна приносить пользу пользователю или бизнесу.

Estimable — оцениваемая.

Команда должна понимать объем работы.

Small — небольшая.

История не должна быть слишком большой.

Testable — проверяемая.

Должно быть понятно, как проверить результат.

Роль бизнес-аналитика в работе с User Story

Бизнес-аналитик помогает:

— выявить потребность пользователя;

— определить правильную роль;

— сформулировать ценность;

— уточнить детали;

— выявить ограничения;

— подготовить критерии приемки.

User Story и требования

Важно понимать: User Story не заменяет все требования.

В сложных системах дополнительно могут понадобиться:

— бизнес-правила;

— модели процессов;

— схемы данных;

— ограничения;

— технические требования.

User Story — это отправная точка для понимания потребности.

Acceptance Criteria — критерии приемки

Каждая хорошая User Story должна иметь критерии, по которым можно определить успешность реализации.

Например:

User Story: Как клиент банка, я хочу открыть счет онлайн, чтобы не посещать отделение.

Acceptance Criteria:

— клиент может заполнить заявку через приложение;

— система проверяет данные пользователя;

— после успешной проверки счет создается автоматически;

— клиент получает уведомление.

Формат Given-When-Then

Популярный способ описания сценариев.

Пример:

Given.

Пользователь имеет зарегистрированный аккаунт.

When.

Пользователь выбирает восстановление пароля.

Then.

Система отправляет инструкцию для восстановления доступа.

Практический пример №1

Интернет-магазин.

Бизнес просит: «Добавить фильтры товаров».

Исследую проблему.

Выясняется: пользователи не находят нужные товары.

User Story: Как покупатель интернет-магазина, я хочу быстро фильтровать товары по параметрам, чтобы находить подходящие варианты без просмотра сотен позиций.

Ценность становится понятной.

Практический пример №2

CRM-система.

Запрос: «Добавить новый отчет для отдела продаж».

Аналитик уточняет: Зачем нужен отчет?

Оказывается: менеджеры хотят быстрее видеть клиентов, которым нужно сделать повторный контакт.

User Story: Как, я хочу видеть список клиентов, требующих повторного менеджер продаж контакта, чтобы увеличивать количество успешных сделок.

Практический пример №3

Мобильное приложение.

Запрос: «Добавить уведомления».

Выясняю: Проблема не в отсутствии уведомлений.

Пользователи забывают завершить оплату заказа.

User Story: Как покупатель, я хочу получать напоминание о незавершенном заказе, чтобы не потерять выбранный товар.

Типичные ошибки при создании User Story

Ошибка №1. Описывать техническое решение вместо потребности.

Ошибка №2. Не указывать пользователя.

Ошибка №3. Забывать про ценность.

Ошибка №4. Создавать слишком большие истории.

Ошибка №5. Не определять критерии приемки.

Как думает сильный аналитик

Он понимает: User Story — это не просто удобный формат записи требований. Это способ заставить команду думать о пользователе.

Поэтому аналитик задает вопросы:

— Кто испытывает проблему?

— Какую цель хочет достичь пользователь?

— Почему это важно?

— Как мы поймем, что решение работает?

— Какую ценность получает бизнес?

Чек-лист хорошей User Story

☐ Определен пользователь.

☐ Понятна потребность.

☐ Указана ценность.

☐ История имеет разумный размер.

☐ Есть критерии приемки.

☐ Команда понимает цель изменения.

☐ Решение обсуждается совместно.

Главное правило

Хорошая User Story начинается не с функции, а с потребности человека. Задача бизнес-аналитика — помочь команде создавать не просто набор возможностей, а решения, которые действительно помогают пользователям достигать своих целей и создают ценность для бизнеса.

Вопрос №126. Чем отличаются User Story, Use Case и функциональные требования?

Почему этот вопрос важен

Бизнес-аналитики используют разные способы описания требований.

На практике часто возникает ситуация: Один специалист пишет User Story. Другой предпочитает Use Case. Третий создает подробные функциональные требования. Все они могут быть правильными.

Но проблема возникает тогда, когда команда не понимает:

— какой формат использовать;

— какую информацию он должен содержать;

— когда один подход лучше другого.

Сильный аналитик должен уметь выбирать подходящий инструмент под конкретную задачу.

Главная идея вопроса

User Story описывает потребность пользователя и ценность. Use Case подробно показывает взаимодействие пользователя с системой. Функциональные требования фиксируют, что именно должна делать система.

1. ***Что такое User Story.***

User Story — это краткое описание потребности пользо-

вателя.

Она отвечает на вопрос: «Кто хочет что-то сделать и зачем это нужно?»

Формат:

Как [роль], я хочу [действие], чтобы [ценность].

Пример:

Как клиент интернет-магазина, я хочу сохранить товары в избранное, чтобы быстро вернуться к ним позже.

Что мы понимаем:

— кто пользователь;

— какая потребность;

— какая ценность.

Но мы еще не знаем:

— какие экраны будут;

— какие шаги выполняет система;

— какие ошибки возможны.

Когда использовать User Story

Лучше всего подходит для:

— Agile-команд;

— продуктовой разработки;

— Scrum;

— быстрых изменений;

— работы с Backlog.

Главная цель: понять потребность пользователя.

2. ***Что такое Use Case.***

Use Case (вариант использования) — это подробное опи-

сание взаимодействия пользователя и системы для достижения определенной цели.

Он отвечает на вопрос: «Как пользователь и система взаимодействуют для выполнения задачи?»

Пример:

Use Case:

Название: Оформление заказа.

Актор: Покупатель.

Основной сценарий:

— Пользователь открывает корзину.

— Система показывает выбранные товары.

— Пользователь вводит адрес доставки.

— Система рассчитывает стоимость.

— Пользователь выбирает способ оплаты.

— Система создает заказ.

— Пользователь получает подтверждение.

Что обычно содержит Use Case

— название сценария;

— участники;

— цель;

— предварительные условия;

— основной поток действий;

— альтернативные сценарии;

— исключения;

— результат.

Когда использовать Use Case

Подходит для:

— сложных систем;

— банковских решений;

— государственных систем;

— корпоративных приложений;

— процессов с большим количеством сценариев.

Главная цель: понять последовательность взаимодействия пользователя и системы.

3. **Что такое функциональные требования**

Функциональные требования описывают конкретные возможности системы.

Они отвечают на вопрос: «Что система должна делать?»

Примеры:

— система должна создавать учетную запись пользователя;

— система должна отправлять уведомление после оплаты;

— система должна рассчитывать стоимость доставки;

— система должна блокировать аккаунт после пяти неудачных попыток входа.

Когда использовать функциональные требования

Подход подходит для:

— технически сложных проектов;

— системной разработки;

— интеграций;

— формальных требований;

— проектов с жестким контролем документации.

Главная цель: зафиксировать ожидаемое поведение системы.

Основные различия

Подход	Главный вопрос	Фокус
User Story	Что хочет пользователь и зачем?	Пользовательская ценность
Use Case	Как пользователь взаимодействует с системой?	Сценарий работы
Функциональные требования	Что должна делать система?	Возможности системы

Пример на одной задаче.

Компания создает мобильный банк.

User Story.

Как клиент банка, я хочу переводить деньги через приложение, чтобы быстро отправлять средства другим людям.

Фокус: потребность клиента.

Use Case.

Название: Перевод денежных средств.

Сценарий:

— Клиент открывает раздел переводов.

— Выбирает получателя.

— Вводит сумму.

— Подтверждает операцию.

— Система проверяет данные.

— Система выполняет перевод.

Фокус: взаимодействие.

Функциональные требования.

— Система должна поддерживать перевод между счетами.

— Система должна проверять баланс клиента.

— Система должна отправлять уведомление после успешной операции.

Фокус: функциональность.

Как они могут использоваться вместе

В зрелых командах эти подходы не заменяют друг друга. Они дополняют.

Например:

User Story:
Как клиент, я хочу оплатить покупку онлайн.
↓
Use Case:
Подробное описание процесса оплаты.
↓
Функциональные требования:
Описание возможностей платежной системы.

Роль бизнес-аналитика

Аналитик должен понимать: какой уровень детализации нужен команде.

Для обсуждения идеи достаточно User Story.

Для сложного процесса нужен Use Case.

Для разработки системы необходимы функциональные требования.

Практический пример №1

Простое изменение.

Задача: Добавить возможность смены фотографии про-

филя.

Достаточно: User Story:

Как пользователь, я хочу менять фотографию профиля, чтобы персонализировать свой аккаунт.

Практический пример №2

Сложный процесс.

Задача: Оформление ипотечного кредита. Одной User Story недостаточно.

Необходимо описать:

— участников процесса;

— проверки;

— документы;

— исключения;

— взаимодействие систем.

Здесь лучше использовать Use Case.

Практический пример №3

Интеграция систем.

Задача: Обмен данными между CRM и ERP.

Здесь потребуются функциональные требования:

— какие данные передаются;

— когда происходит обмен;

— какие ошибки возможны;

— как система должна реагировать.

Типичные ошибки аналитиков

Ошибка №1. Использовать User Story для описания сложной системы без детализации.

Ошибка №2. Создавать огромные документы там, где достаточно простой User Story.

Ошибка №3. Описывать решение вместо потребности пользователя.

Ошибка №4. Путать бизнес-требования и функциональные требования.

Ошибка №5. Выбирать формат документации без учета особенностей проекта.

Как думает сильный аналитик

Он не спрашивает: «Какой формат требований самый правильный?»

Он спрашивает: «Какой формат лучше поможет команде понять проблему и создать правильное решение?»

Чек-лист выбора подхода

☐ Нужно понять потребность пользователя → User Story.

☐ Нужно описать взаимодействие пользователя и системы → Use Case.

☐ Нужно зафиксировать поведение системы → функциональные требования.

☐ Уровень детализации соответствует сложности проекта.

☐ Команда понимает назначение документа.

Главное правило

User Story показывает, какую ценность хочет получить

пользователь. Use Case объясняет, как пользователь взаимодействует с системой. Функциональные требования фиксируют, что именно система должна делать. Профессиональный бизнес-аналитик умеет использовать все три подхода и выбирает инструмент, который лучше всего помогает команде создавать ценное решение.

Вопрос №127. Как бизнес-аналитику управлять изменениями требований в проекте?

Почему этот вопрос важен

Практически ни один проект не проходит полностью по первоначальному плану.

В процессе работы появляются новые обстоятельства:

— изменились требования бизнеса;

— пользователи предложили новые идеи;

— обнаружились ограничения системы;

— изменилось законодательство;

— появились новые риски;

— руководство пересмотрело приоритеты.

Некоторые команды воспринимают изменения как проблему. Но профессиональный аналитик понимает: изменения — это естественная часть любого проекта. Главная задача не в том, чтобы полностью исключить изменения, а в том, чтобы управлять ими правильно.

Главная идея вопроса

Управление изменениями требований — это процесс оценки, согласования, документирования и контроля изменений, чтобы сохранить баланс между бизнес-ценностью, сроками, стоимостью и качеством проекта.

Почему требования меняются

Причины изменений могут быть разными.

1. ***Изменение бизнес-потребностей.***

Компания может изменить стратегию.

Например: Изначально компания хотела увеличить количество клиентов.

Через месяц появилась новая цель: снизить операционные расходы.

Требования к системе могут измениться.

2. ***Новая информация.***

Во время анализа могут обнаружиться факты, которых команда не знала раньше.

Например: После интервью с пользователями выяснилось, что проблема находится не там, где предполагалось.

3. ***Изменения внешней среды.***

Например:

— новые законы;

— действия конкурентов;

— изменение рынка;

— новые технологии.

4. ***Ошибки первоначального анализа.***

Иногда требования меняются потому, что первоначально проблема была понята неправильно. Это не всегда ошибка. Часто именно в процессе работы появляется более глубокое понимание ситуации.

Почему нельзя просто принимать все изменения

Представим проект разработки интернет-магазина. Заказчик говорит: «Добавим еще возможность оплаты через десять новых сервисов». Кажется, изменение полезное.

Но аналитик должен оценить:

— сколько займет разработка;

— какие системы затронуты;

— изменится ли срок запуска;

— какие риски появятся.

Каждое изменение имеет последствия.

Процесс управления изменениями

Шаг 1. Регистрация изменения.

Любое изменение должно быть зафиксировано.

Например:

— описание изменения;

— инициатор;

— причина;

— ожидаемый результат.

Шаг 2. Анализ влияния изменения.

Это одна из ключевых задач аналитика.

Необходимо оценить:

Влияние на бизнес.

— увеличит ли изменение ценность;

— решает ли оно реальную проблему.

Влияние на требования.

— какие требования изменятся;

— какие новые требования появятся.

Влияние на систему.

— какие компоненты затронуты;

— нужны ли новые интеграции.

Влияние на проект.

— изменятся ли сроки;

— потребуется ли дополнительный бюджет;

— нужны ли новые ресурсы.

Шаг 3. Подготовка вариантов решения.

Аналитик не должен просто сказать: «Это невозможно».

Его задача — предложить варианты.

Например:

Вариант 1: реализовать изменение сейчас.

Вариант 2: включить его в следующий релиз.

Вариант 3: найти более простой способ достижения цели.

Шаг 4. Согласование решения

Изменение должно быть принято ответственными лицами.

В зависимости от проекта это могут быть:

— заказчик;

— Product Owner;

— руководитель проекта;

— комитет изменений.

Шаг 5. Обновление документации.

После согласования необходимо обновить:

— требования;

— User Stories;
— модели процессов;
— спецификации;
— планы проекта.

Шаг 6. Контроль реализации.

После принятия изменения аналитик проверяет:
— реализовано ли оно правильно;
— достигнут ли ожидаемый эффект;
— не появились ли новые проблемы.

Матрица оценки изменений

Удобно оценивать изменения по двум параметрам:

Ценность.

Насколько изменение важно для бизнеса.

Стоимость.

Сколько ресурсов потребуется.

Высокая ценность + низкая стоимость: → реализовать быстро.

Высокая ценность + высокая стоимость: → планировать отдельно.

Низкая ценность + высокая стоимость: → пересмотреть необходимость.

Практический пример №1

CRM-система.

Компания внедряет CRM. После начала проекта отдел продаж просит добавить автоматические рекомендации клиентам.

Оцениваю:
Плюсы:
— потенциальный рост продаж.

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.