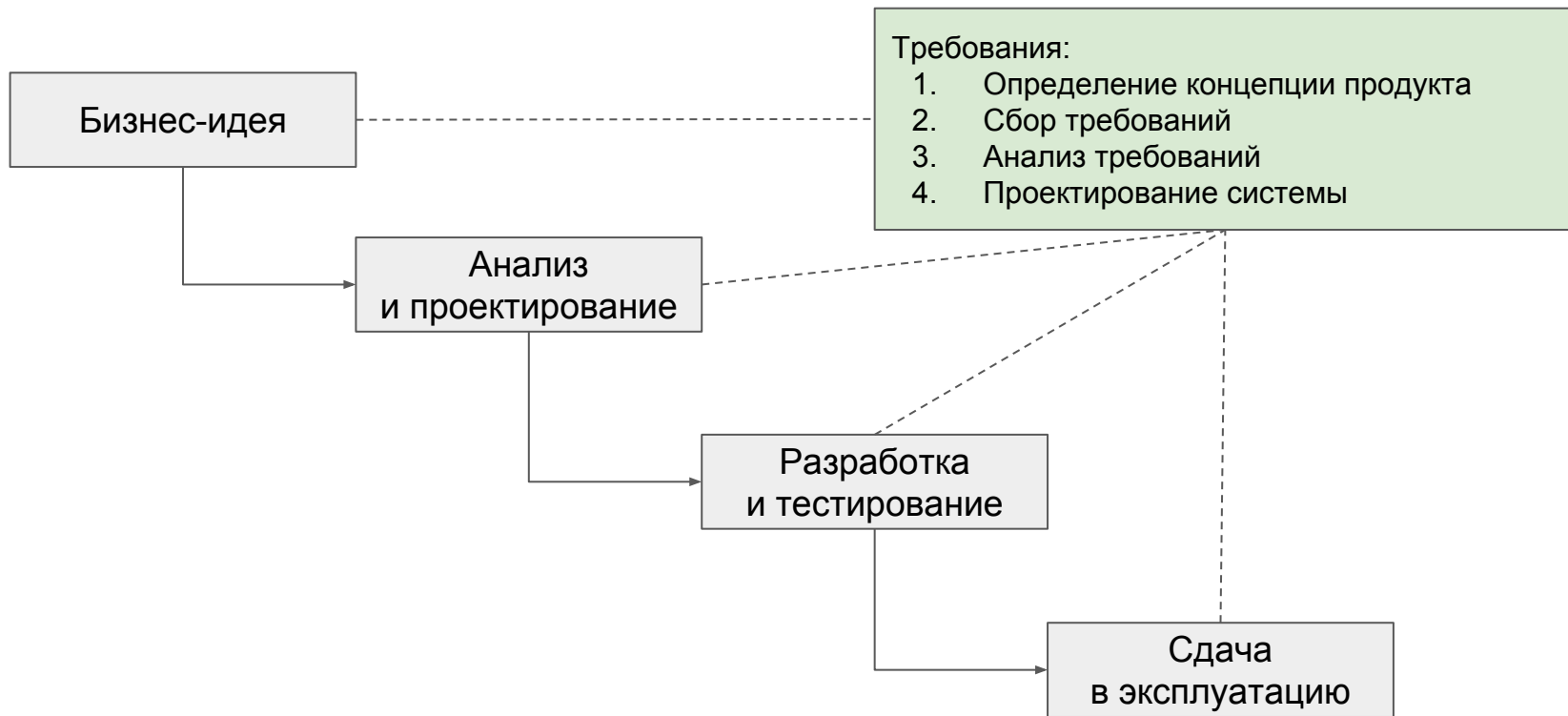


Бизнес-анализ

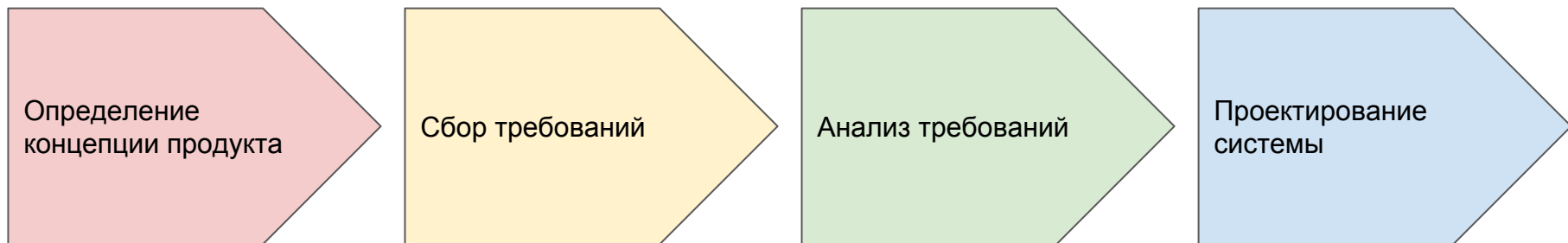
Тема 3

Деятельность бизнес-аналитика

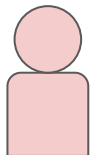
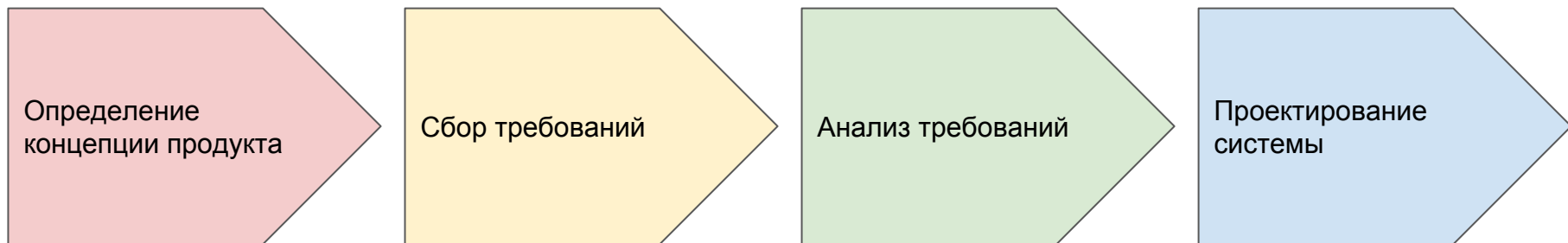
Активность бизнес-аналитика



Этапы сбора и анализа требований

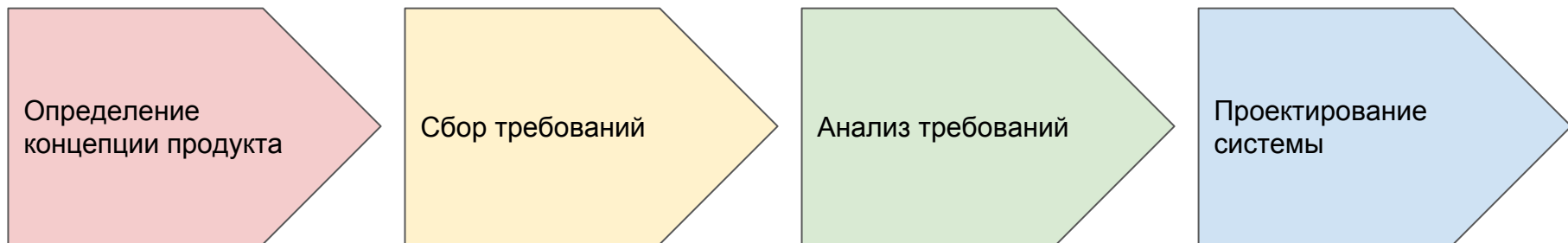


Этапы сбора и анализа требований



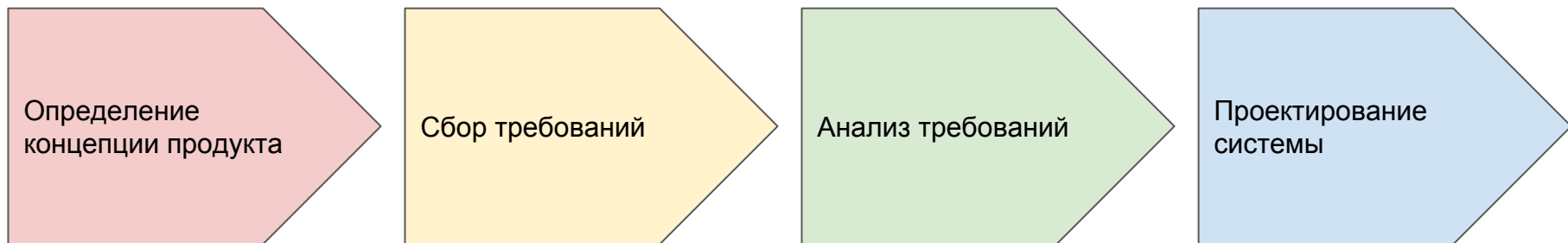
- Работа с инвестором
- Выработка видения продукта
- Результат: Решение о запуске (незапуске) продукта

Этапы сбора и анализа требований



- Работа с заказчиком и будущими пользователями
- Определение наиболее важных функций продукта для построения MVP

Этапы сбора и анализа требований

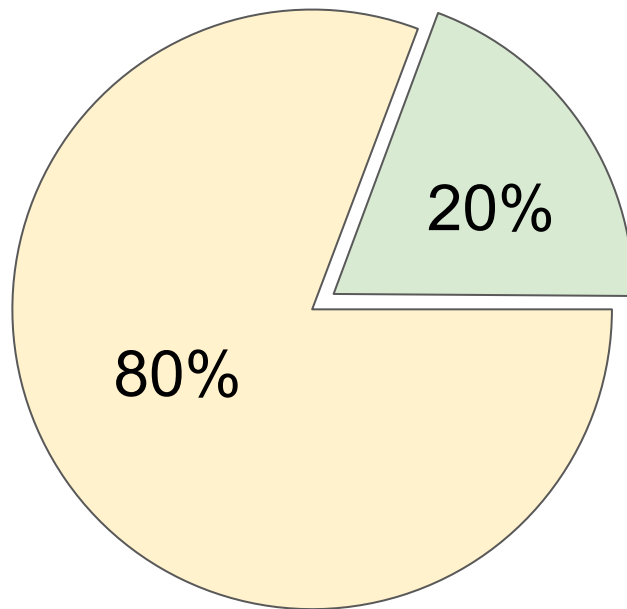


- Структуризация собранных требований
- Цель: Четкий список не дублируемых требований, чтобы обойтись минимальным функционалом и не выйти за рамки бюджета/сроков



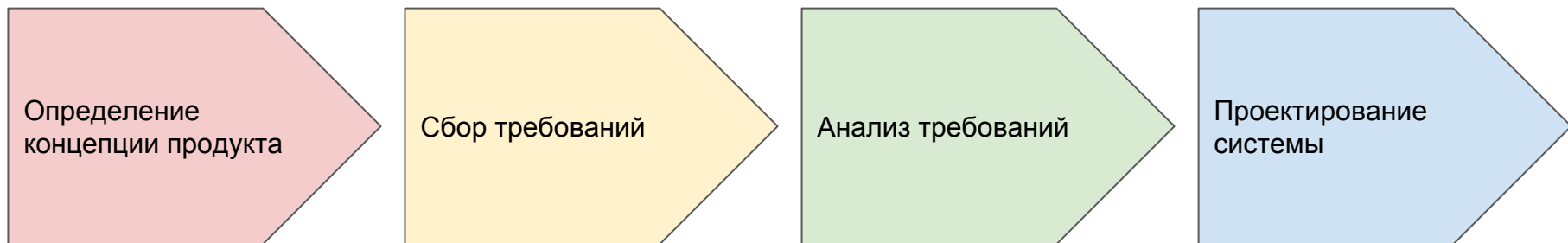
Принцип Парето (для ИТ)

Делать только те 20% функционала, которые обеспечивают 80% сценариев использования *



* В продукте должна быть минимум один (ключевой) функционал, качеством которого пренебрегать нельзя

Этапы сбора и анализа требований



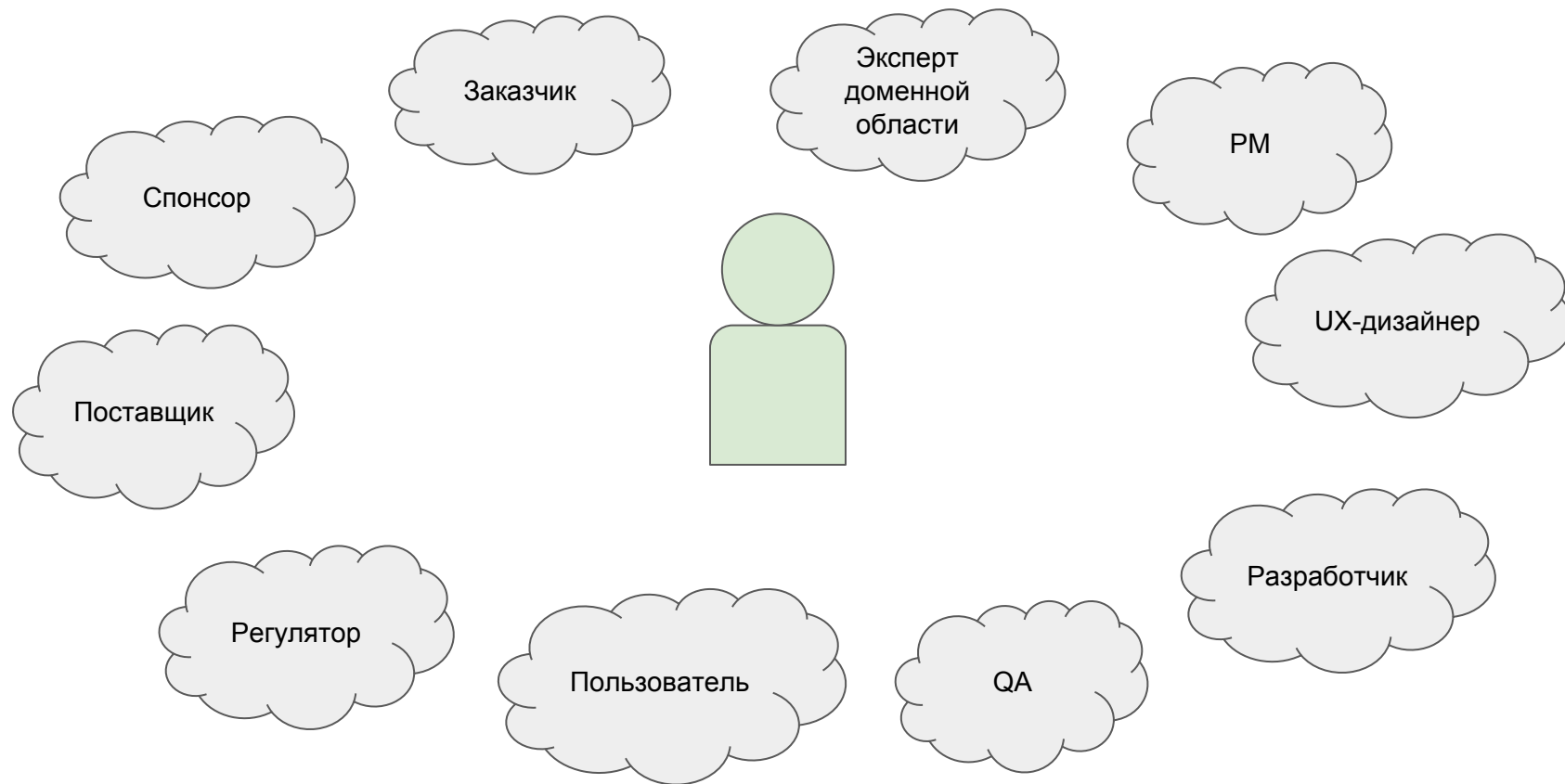
- Принятие проектных решений о функционале продукта
- Результат: Техническое задание к продукту, которое содержит полное описание поведения продукта



Принципы бизнес-анализа

Ищем коренные причины	А не исследуем симптомы
Улучшаем бизнес	А не внедряем ИТ систему
Креатив	А не шаблонные решения
Реализуемые требования	А не все возможные предложения
Полный жизненный цикл	А не только фаза проектирования
Переговоры	А не избегание конфликта
Скорость развертывания	А не стремление к совершенству

Бизнес-аналитик и другие стейкхолдеры



Основная деятельность БА

- Активно **взаимодействует с заказчиком**, превращает “поток сознания” заказчика в формализованные требования (понятные, полные, без противоречий)
- Если заказчик “не знает, чего хочет”, то **предлагает оптимальное решение** или “подводит” к нему
- **Анализирует влияние** новых требований на существующую архитектуру и функционал (общаясь с командой, изучая тех. документацию)
- **Документирует** требования в виде документа в требуемом виде, согласовывает и утверждает их
- **Внедряет требования** в процесс разработки (через тикет-систему), отслеживает активность по ним, отвечает на вопросы
- Осуществляет **верхнеуровневый контроль** соответствия реализованного функционала требованиям
- **Управляет изменениями** требований

Чем еще занимается БА

- Проводит анализ рыночных решений и сравнение конкурентов
- Проактивно предлагает новые “фичи”
- Участвует или сам готовит документацию по продукту
- Участвует в сдаче продукта заказчику
- По мере сил отвечает на технические вопросы конечных пользователей
- Готовит аналитические отчеты (или “достает” необходимую информацию) для руководства
- Берет на себя принятие решений и ответственность
- Является ресурсом, “внешней памятью”, арбитром команды

“Скиллы” БА

Hard skills

- Понимание современных технологий
- Предметная область продукта
- Работа с требованиями
- Бизнес-процессы
- UX
- Навыки решения проблем
- Грамотный язык
- Умение работать с текстами

Soft skills

- Аналитические способности
- Системное мышление
- Коммуникативность
- Ответственность
- Проактивность
- Широкий кругозор

Техники выявления требований

Техники выявления требований

Интервью

Фокус-группы

Мозговой штурм

Семинары

Наблюдение

Моделирование
процессов

Анализ документов

Анализ интерфейсов

Опросы

Прототипы

Варианты
использования

Интервью

- Задавайте вопросы, которые помогут понять в чем дело
 - Формально или неформально
 - Индивид или группа
 - Открытые вопросы устраняют пробелы
 - Закрытые вопросы (Да/Нет) нужны для подтверждения
- От чего зависит успех
 - Степень знакомство между участниками
 - Опыт интервьюера
 - Готовность интервьюируемого делиться информацией
 - Навыки документирования



Интервью

Подготовка

- Согласовать время, место и список участников
 - Добавить в календарь
 - Обозначить тематику и структуру
 - Получить подтверждение
- Изучить релевантные документы
 - Выслать участникам или дать ссылку
- Подготовить список вопросов
 - Выслать участникам

Проведение

- Начало
 - Размещение / рассадка
 - Использование диктофона
 - Завязка
- Вопросы и фиксация ответов
 - Главный вопрос -- зачем?
 - Требуемые возможности + ограничения
 - Хронометраж
- Окончание
 - Какие вопросы были не заданы?
 - Повторить договоренности
 - Обозначить план действий
- После интервью
 - Выслать протокол
 - Начать работу asap

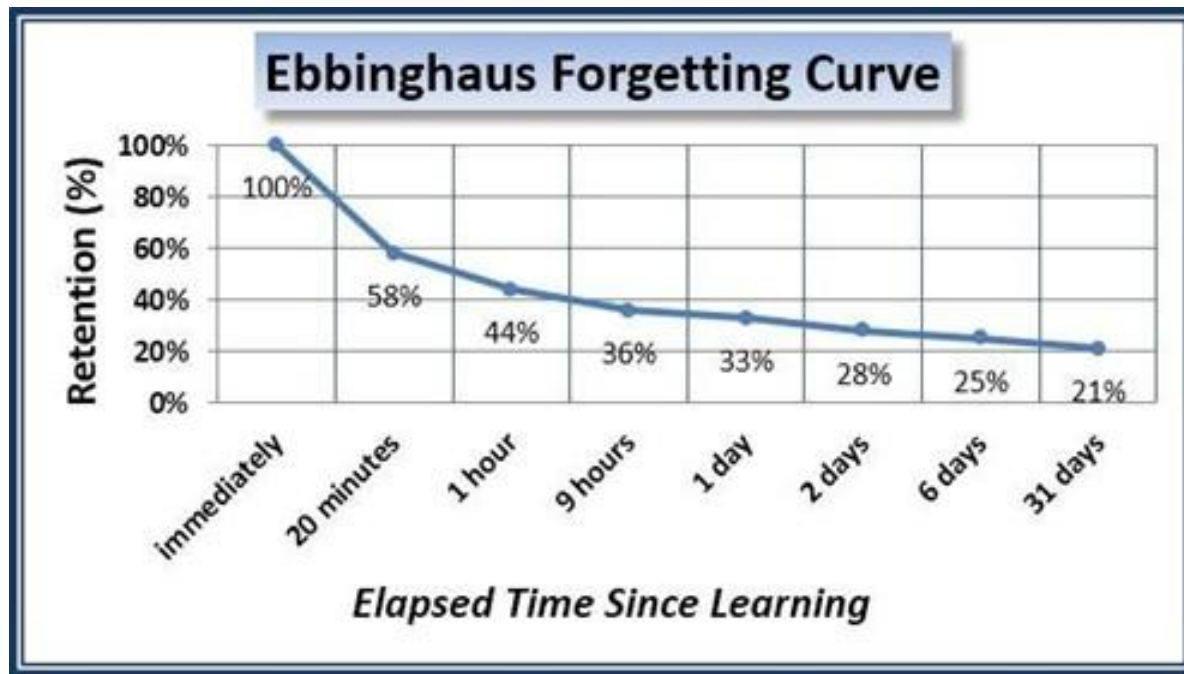
Пример интервью

- Определение профиля заказчика
 - Метаданные
 - Каковы ваши обязанности?
 - Что вы производите? Для кого?
 - Как вы измеряете свой успех?
 - Какие проблемы влияют на успешность?
 - Что делает вашу работу проще? Сложнее?
- Оценка проблемы
 - Для каких проблем у вас нет хороших решений?
 - Почему существует эта проблема?
 - Как она решается сейчас?
 - Как бы вы хотели ее решать?
- Пользователи
 - Кто ваши пользователи?
 - Какое у них образование? Уровень владения компьютером?
 - Какие технологии используются пользователями?
 - Какие ваши ожидания от продукта?
 - Сколько времени нужно пользователям, чтобы освоиться с продуктом?
- Резюме
 - “Итак, вы сказали мне ...”
 - Адекватен ли этот список проблем?
 - Какие еще есть проблемы?

Пример интервью

- Предположения насчет проблем
 - Является ли проблема реальной?
 - Каковы ее причины?
 - Как она решается в настоящее время?
 - Как ее следует решать?
 - Насколько ее решение важно?
- Оценка решения
 - Что если у вас будут следующие возможности?
 - Как вы расцениваете важность ...?
- Оценка возможности
 - Кто нуждается в этом продукте? Сколько таких пользователей?
 - Насколько внедрение продукта значимо?
- Оценка нефункциональных требований
 - Каковы ожидания относительно производительности, надежности?
 - Кто будет заниматься поддержкой продукта?
 - Каковы требования по безопасности?
 - Как продукт должен быть интегрирован?
 - Есть ли зависимости от других систем?
- Другое
 - Есть ли законодательные требования, стандарты, которых нужно придерживаться?
 - Есть ли еще требования, о которых нужно знать?
 - С кем и как лучше связаться, если возникнут новые вопросы?

Кривая забывания Эббингауза



Некоторые хорошие практики

Видеоконференции

- Проще голосом, видео не обязательно
- Переговорка, но не рабочее место
- Много людей -- труднее общаться
- Иметь альтернативный вариант:
 - Skype
 - Hangouts
 - Zoom
 - Citrix
- Расшаривание экрана
- Непустой баланс Skype
- Записывать важные детали

Переписка

- IMAP ящик, доступ отовсюду
- Не писать длинных писем
- Первая строка -- резюме
- Новая тема -- новый тред
- [Inbox Zero](#), ярлыки
- Проблема не убирается пока не решена
- Копии (cc) и скрытые копии (bcc)
- Автоответчик
- Возможность отмены отправления
- Не писать ничего компрометирующего
- Отвечать как можно быстрее, особенно если ASAP

Фокус-группы

- Опрос группы модератором
 - Формальный процесс
 - Структурированный процесс
 - 6-12 участников
- Роль модератора
 - Поддерживает общение
 - Задает открытые вопросы
 - Вовлекает в общение всех участников
 - Остается нейтральным
- Можно опросить несколько человек за один раз



[Сцена с фокус-группой](#)
из 2-го сезона “Кремниевой долины”

Недостатки группового обсуждения

- Отнимает время сразу у многих участников
- Часть людей может вообще не прийти
- Концентрация на личных амбициях, а не на решении проблем
- Политические игры
- Страх показаться некомпетентным
- Страх перед присутствующим руководством

Мозговой штурм

- Временной лимит
- Визуализация
- Ведущий
- Небольшой размер группы (6-8 чел.)
 - Разные по уровню и областям знаний
- Общие правила:
 - Критерии оценки и ранжирования идей
 - Без критики
 - Без лишних обсуждений
- В конце произвести оценку идей



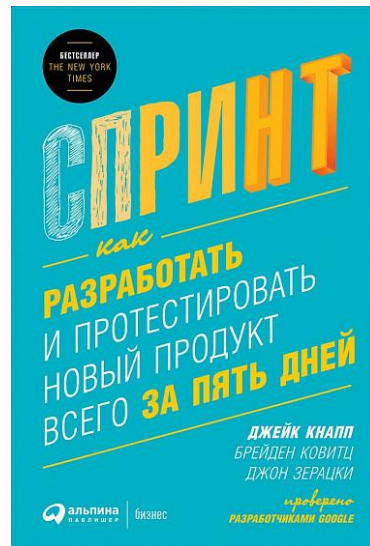
Семинары (workshops)

- Один из лучших (?) методов
 - Есть программа
 - Тщательный выбор участников
 - Подготовительные материалы
 - Опытный, нейтральный ведущий
 - Запись ведется секретарем
 - 1-2 дня
- Участники
 - Представители всех заинтересованных сторон
 - Большое количество участников замедляет процесс
 - Небольшое количество -- создает пробелы



Семинары (workshops)

- Проведение семинара
 - Повестка, правила
 - Текущее состояние проекта, цели семинара
 - Мозговой штурм
 - Определение возможностей и ограничений
 - Определение приоритетов и фаз проекта
 - Определение дальнейших шагов



Пятидневный вариант воркшопа описан в книге
[“Спринт. Как разработать и протестировать новый продукт всего за пять дней”](#)

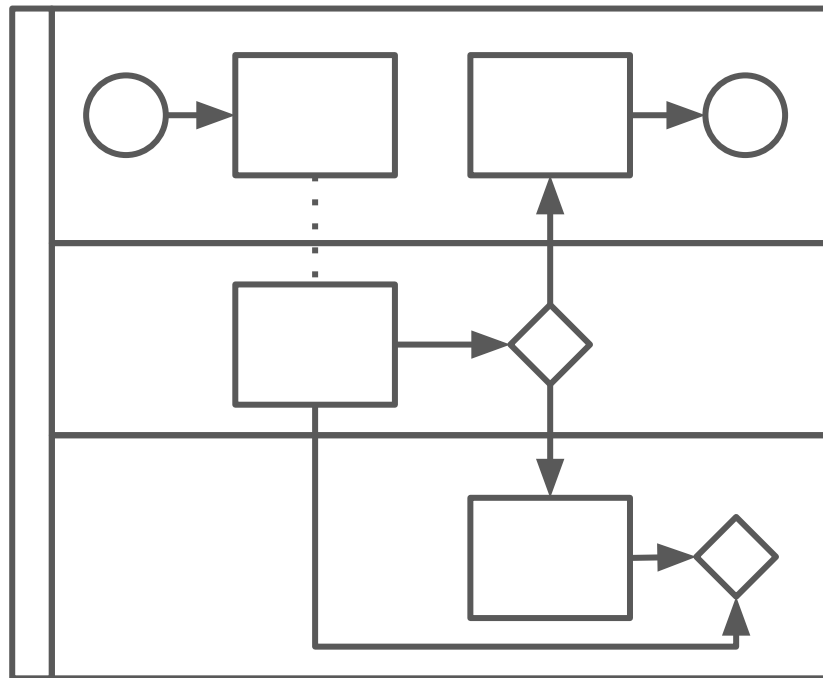
Наблюдение

- Изучите методы работы стейкхолдеров
 - Документирование процесса
 - Режим “невидимки”
 - Не задает вопросов
 - Только записывает
 - Активный режим
 - Диалог с пользователем
 - Realtime
- Может быть затратным по времени
- Можно не увидеть всех (важных) сценариев работы



Моделирование процессов

- Процессный подход
 - Последовательность действий по преобразованию входов в выходы
- Процесс
 - Цель
 - Границы
 - Владелец
 - Функции
 - Ресурсы
- Методологии моделирования
 - Блок-схема
 - Диаграмма активности
 - EPC
 - BPMN

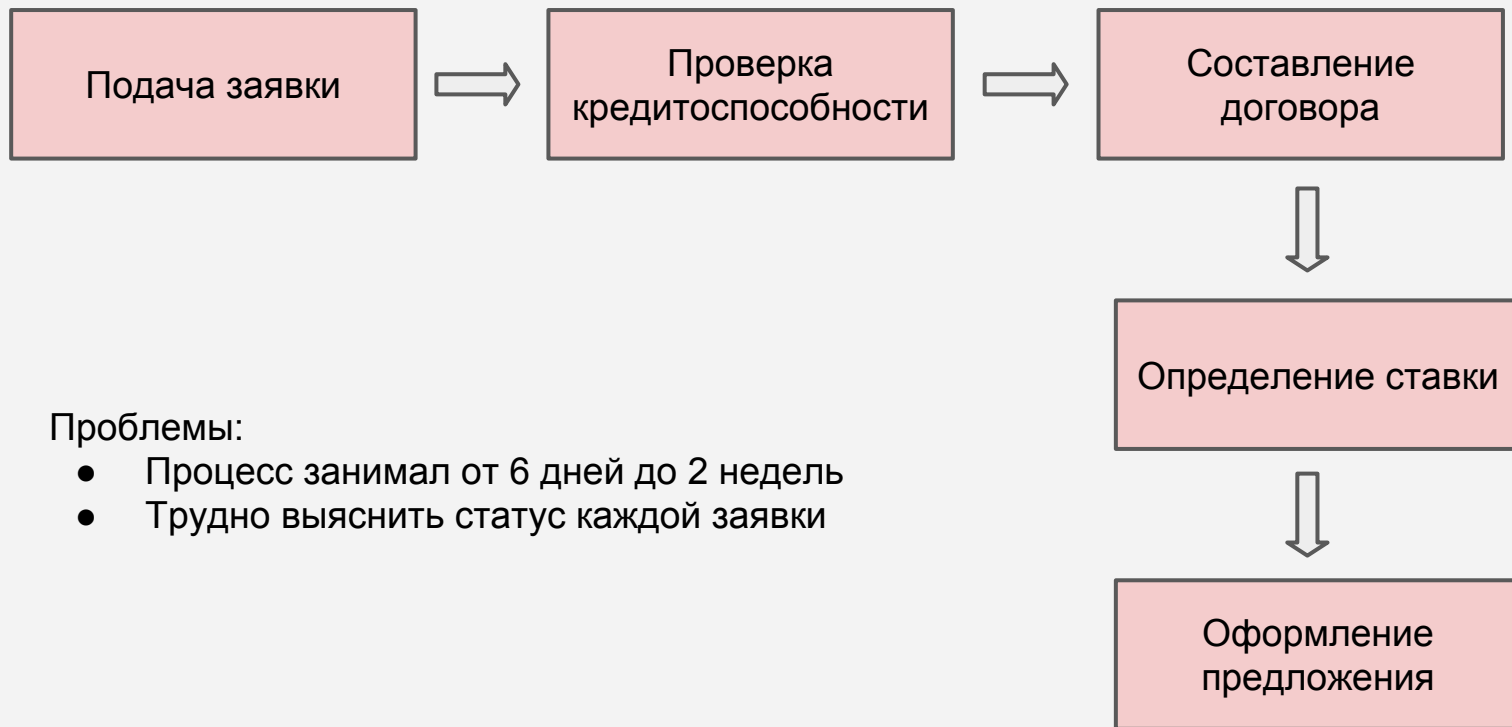


Пример: IBM Credit

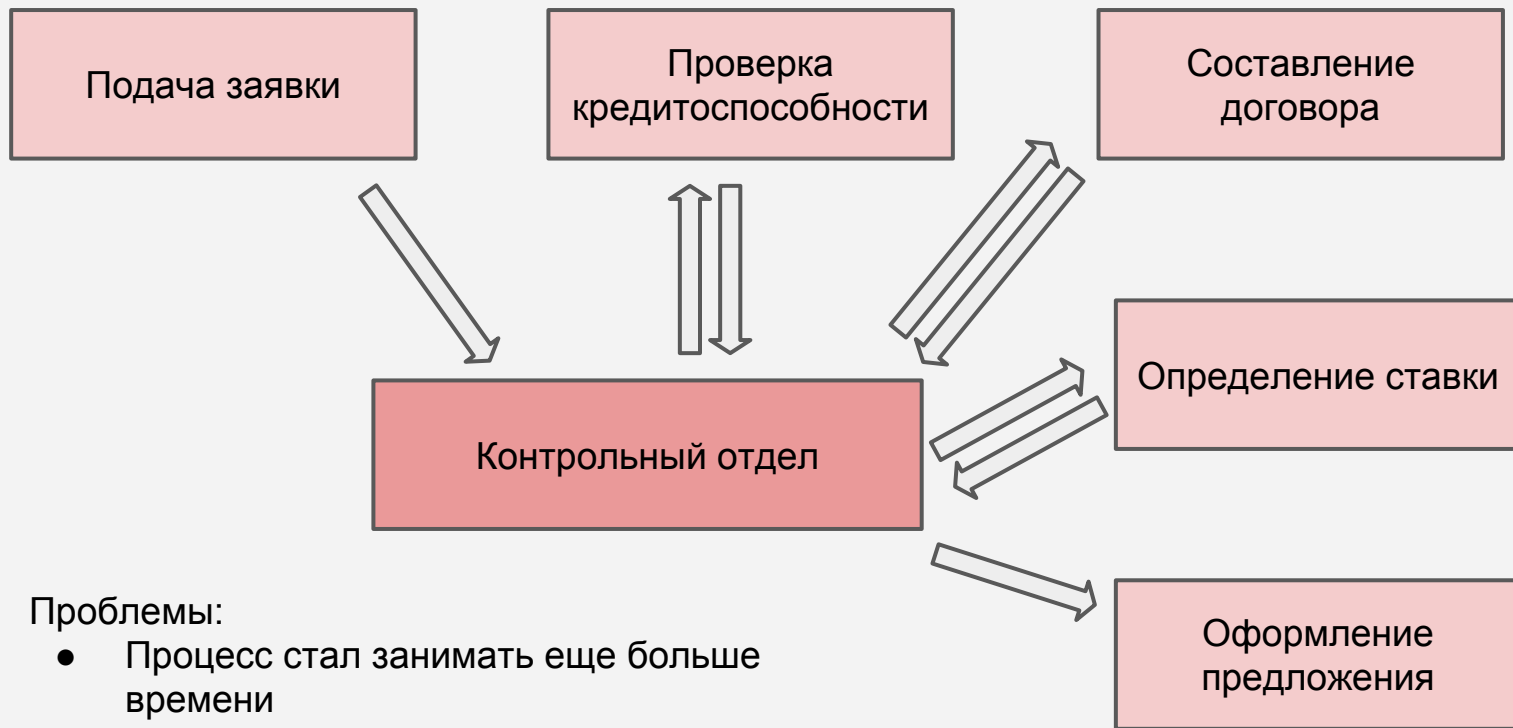
- Кейс в упрощенном виде был описан в книге М. Хаммера и Дж. Чампи “Реинжиниринг корпорации” (1994) и является классическим описанием реинжиниринга процессов
- [Описание кейса](#) по-русски



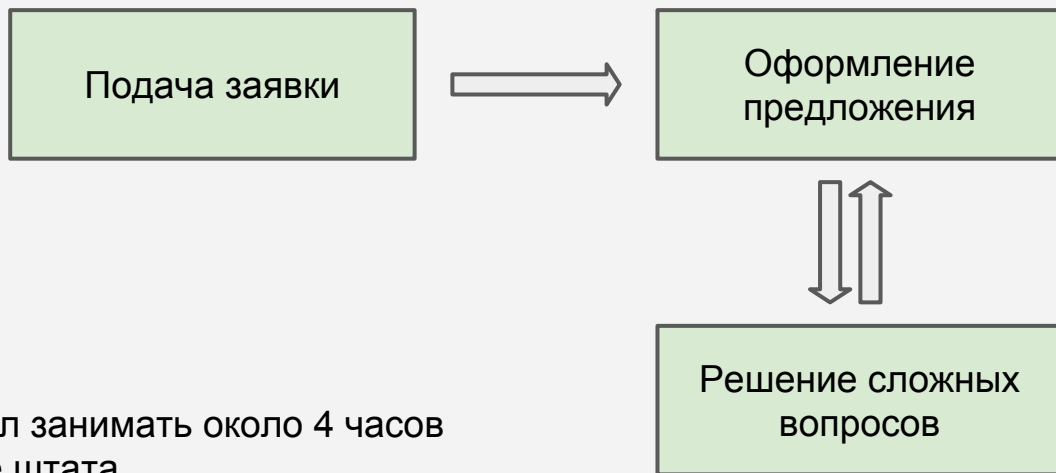
Пример: IBM Credit



Пример: IBM Credit



Пример: IBM Credit



Результаты:

- Процесс стал занимать около 4 часов
- Сокращение штата
- Увеличение производительности в 100 раз

Анализ документов

- Документация
 - Существующие, аналогичные или конкурирующие системы
 - 3rd party системы
 - Переписка
- Аналитика
 - Обзоры, рейтинги, аналитические отчеты и сравнения
- Стандарты
 - Корпоративные, отраслевые, государственные, международные
 - Регламенты, должностные инструкции
 - Отраслевые стандарты, ГОСТы
- Обращения в службу поддержки



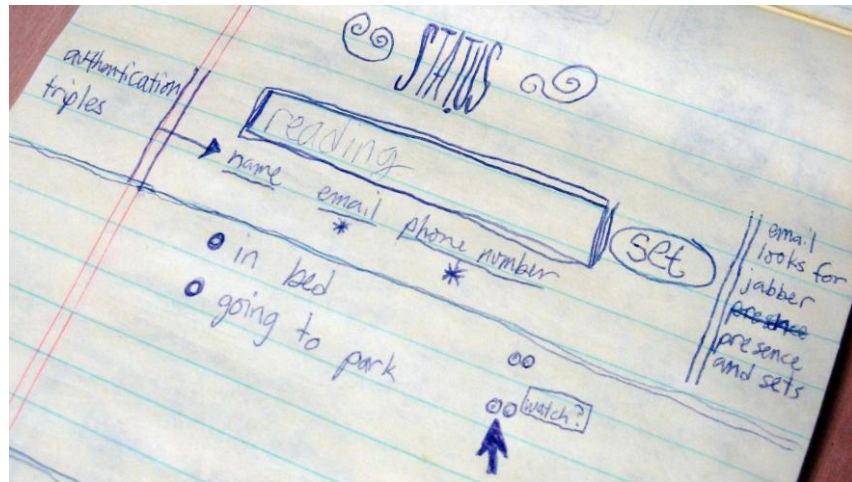
Опросы

- Эффективное получение информации от многих людей
 - Аудитория соответствует цели
 - Временной интервал
 - Простые и понятные вопросы
- Закрытые вопросы
 - Автоматизация полученных результатов
- Открытые вопросы
 - Можно узнать новое, прояснить понимание проблемы
 - Трудно автоматизировать
- Важно участие руководителей



Прототипы

- Наглядность и понятность
- Фокус на конкретных характеристиках системы
- Должен демонстрировать реализацию идей и провоцировать обсуждение
- Не должен быть “слишком крутым”
- Статические и динамические
 - HTML, Bootstrap
 - Balsamiq, Visio, Excel
 - Axure
- Может занять уйму времени, если сосредоточиться на “как” вместо “что”



Один из первых прототипов Twitter

Методология Nota.media

Требования -- это спецификация того, что должно быть реализовано.

В них описано поведение системы, свойства системы или ее атрибуты.

Они могут служить ограничениями в процессе разработки системы.

Ian Sommerville & Pete Sawyer, 1997



Фраза

“А давайте работать без ТЗ?”

открывает дверь в ад

Необходимость и важность требований



Разработка ПО -- это процесс трансформирования чьего-то *замысла* в продукт, удовлетворяющий этому замыслу

Написание требований -- это *процесс выяснения* того, что было задумано

Если вы говорите о том, что вам нужно, вы, *вероятно*, это получите.

Если не говорите, то, *вероятнее всего*, не получите.

Необходимость и важность требований



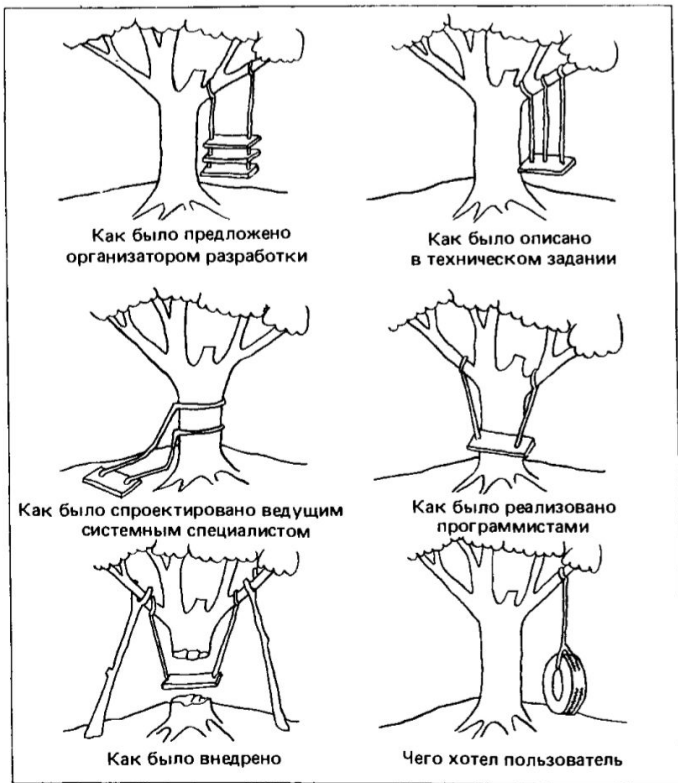
Только около трети всех ИТ-проектов являются успешными

Одной из важнейших причин неуспеха являются плохо написанные требования и их неконтролируемые изменения

Ошибки в требованиях, которые не предотвращены или устранены, попадают в прототипы, код, руководства

Эти ошибки являются самыми дорогими, их труднее всего исправить в дальнейшем

Необходимость и важность требований



Плохо написанные требования делают невозможным точное планирование проекта

Неточные формулировки в требованиях приводят к ненужной работе или переделкам

“Золочение” продукта добавляет ненужные фичи и усложняет продукт

Неполнота требований приводит к неучету потребностей ключевых пользователей

Неустранение проблем с требованиями ведет к ухудшению качества продукта в целом



Задача аналитика:
избежать незапланированных
запросов на изменение
функциональности
(CR, Change Request)

Необходимость и важность требований

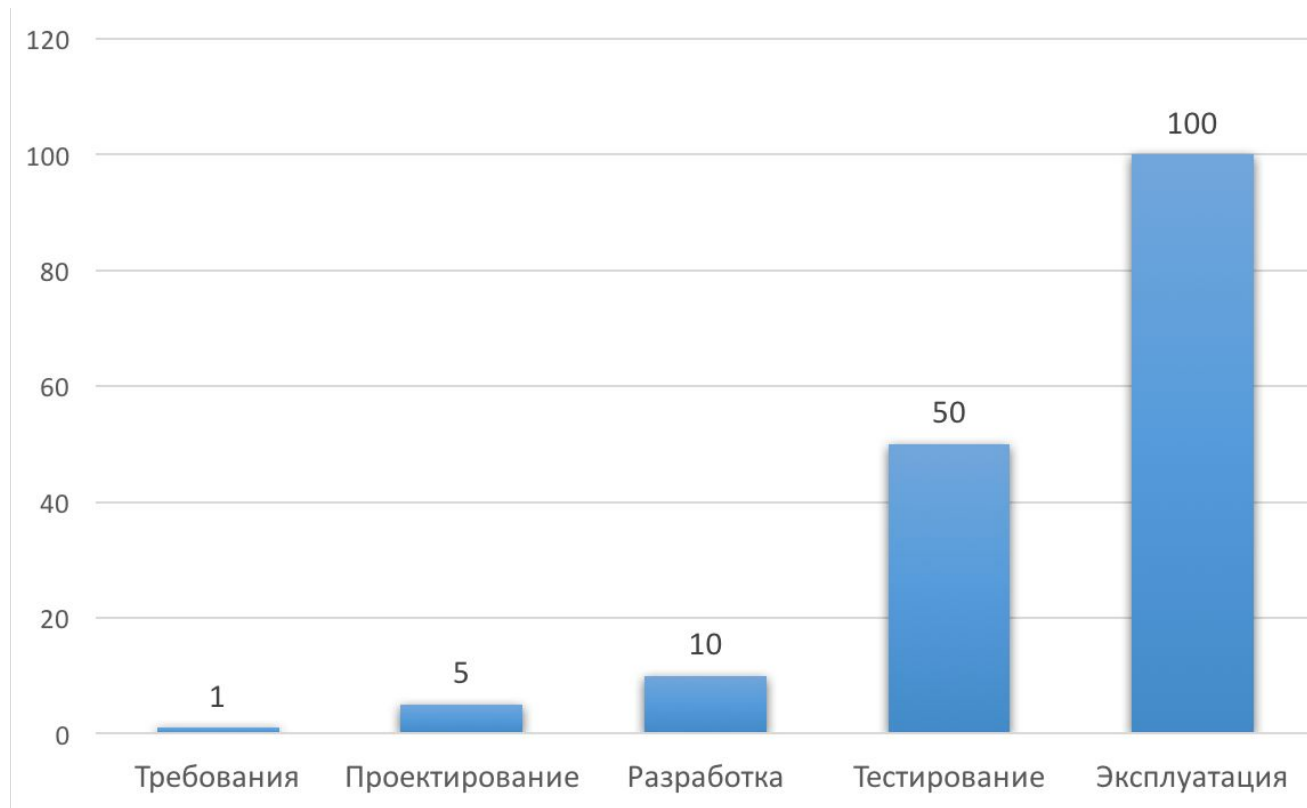
“The hardest single part of building a software system is *deciding precisely what to build*. No other part of the conceptual work is as difficult as establishing the ***detailed technical requirements***, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part of the system is more difficult to rectify later.”

Из статьи Ф. Брукса “Серебряной пули нет”



Фредерик Брукс,
автор “Мифического человеко-месяца”
и “Серебряной пули нет”

Необходимость и важность требований



Основные задачи ТЗ

- Аккумулирует принятые идеи
- Документ, ориентированный на разработчика
- Документ для заказчика
- Юридический документ

Основная задача ТЗ —
помочь создать
качественный продукт

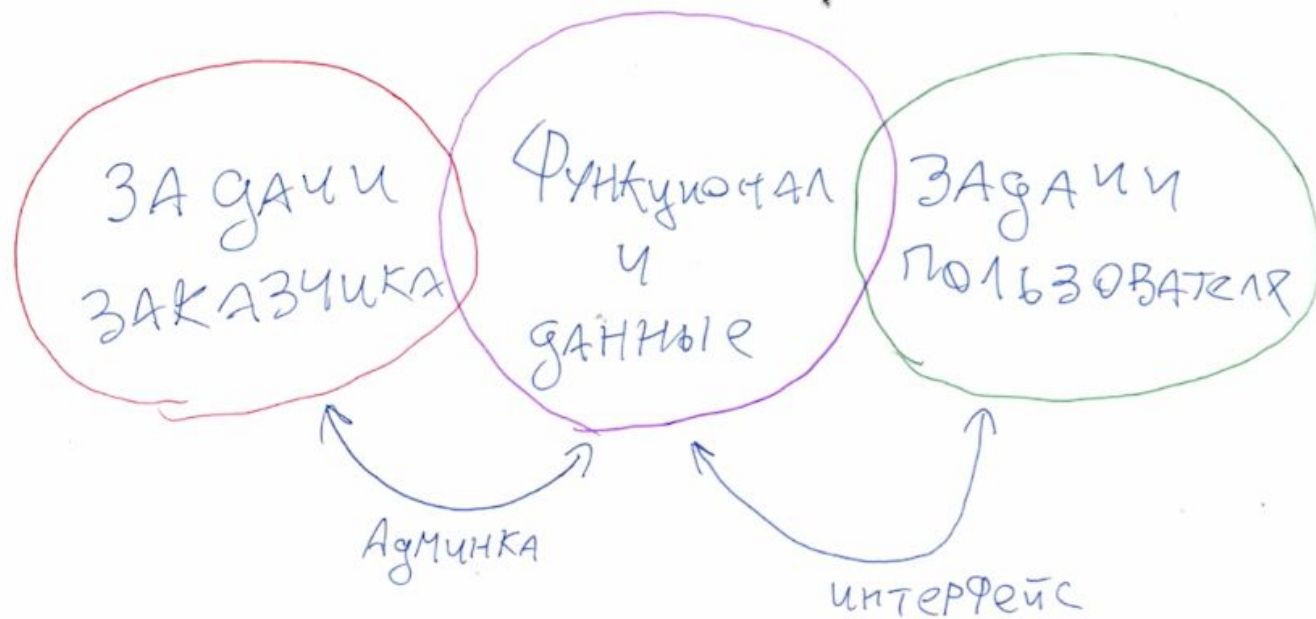
Качества хорошего ТЗ

Однозначность	Не содержит неясных моментов, двусмысленностей
Отчуждаемость	Позволяет работать над продуктом без аналитика :)
Полнота	Содержит всю необходимую информацию
Системность	Обзор разных аспектов системы, учет взаимосвязей
Опрятность	Хороший язык, структура

Требования

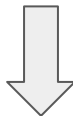
Пользовательские требования	Каким пользовательским целям (потребностям) должен удовлетворять продукт
Функциональные требования	Какой функциональностью должен обладать продукт, чтобы соответствовать пользовательским требованиям
Нефункциональные требования	Каким образом должен функционировать продукт, чтобы обеспечивать качество (юзабилити, производительность, конфиденциальность, безопасность)
Бизнес-правила	Условия, ограничения, формулы, которые будут применяться при работе с системой

С чего начинать писать ТЗ?

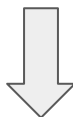


Концепт и пользовательские сценарии

Целевая аудитория	Задачи	Функционал	Страницы
-------------------	--------	------------	----------



Результат: согласованные бизнес-задачи



Далее: пользовательские портреты
(контекст, цели, ограничения, задачи)

“Бумажный тигр” = Архитектура

- Схематическое описание архитектуры решения
 - Структура системы
 - Интерфейсы
 - Логические схемы
 - Структура данных
 - Связь с внешними ресурсами
- “Бумажный тигр”
 - Имеет простой вид
 - Печатается на бумаге
 - Согласовывается с заказчиком

Прототип

- Прототип не должен быть слишком красивым
- Лучше если он будет интерактивным (Axure)
- Опрятность!
- Не должен ограничивать дизайнера

Методология Notamedia

1. Уяснение целей, пользовательских сценариев, функций
 - Результат: Концепт
2. Описание пользовательских сценариев глазами простых пользователей
 - Результат: Пользовательские сценарии
3. Выработка базовой архитектуры системы (структура системы, интерфейсы, логические схемы, структура данных, связь с внешними ресурсами)
 - Результат: “Бумажный тигр”
4. Проработка интерфейсов
 - Результат: Прототип
5. Написание ТЗ
 - Результат: ТЗ

Кто может писать ТЗ

Очень редкая
птица

Должность	БА	UX	Архитектура	Прототипы	ТЗ
Аналитик-проектировщик	+	+	+	+	+
Аналитик	+	+	-	-	-
Проектировщик	-	-	+	+	+
Бизнес-аналитик	+	-	-	-	-
UX-аналитик	-	+	-	-	-
Инф. Архитектор	-	-	+	-	-
Прототипист	-	-	-	+	-
Тех. писатель	-	-	-	-	+

Качества хорошего ТЗ (еще раз)

Однозначность	Не содержит неясных моментов, двусмысленностей
Отчуждаемость	Позволяет работать над продуктом без аналитика :)
Полнота	Содержит всю необходимую информацию
Системность	Обзор разных аспектов системы, учет взаимосвязей
Опрятность	Хороший язык, структура

Структура хорошего ТЗ (начало)

- Общие положения
 - Назначение документа
 - Структура документа
 - Используемые термины
- Технические требования
 - Общие требования = глобальные
 - Требования к системе = серверное оборудование, нагрузка
 - Требования к верстке = стандарты
 - Требования к безопасности = устойчивость, государство
 - Прочие требования = интеграция с внешней системой
- ...

Структура хорошего ТЗ (окончание)

- ...
- Идеология продукта = задачи, аудитория
- Архитектура системы = интерфейсы
- Шаблоны интерфейсов
- Функциональные сценарии
- События и бизнес-правила
- Структура данных
- Справочники
- Протоколы связи с внешними системами
- Требования к системе администрирования = уже будет какая-то CMS, поэтому только функционал

В большом документе важна
системность

Факторы риска для ТЗ

- Начали писать ТЗ очень рано
- Забыли интерфейсы
- Большая простыня текста
- Бессистемность
- Логические дыры
- Слишком простое или наоборот слишком сложное
- ТЗ пишет кто попало (не клиент, не менеджер)
- ~~ТЗ не нравится~~ Это не аргумент, так как ТЗ прежде всего инструмент

Сдача ТЗ

- Едем к клиенту, показываем, объясняем
- Полезно показывать ТЗ по частям
- Полезно показывать ТЗ разработчикам
- Проектировщик продолжает играть много ролей:
 - Консультирует разработчиков
 - Контролирует соответствие сделанного ТЗ
 - Разрешает спорные ситуации
 - Является актуальным источником данных для всех

Государственные стандарты

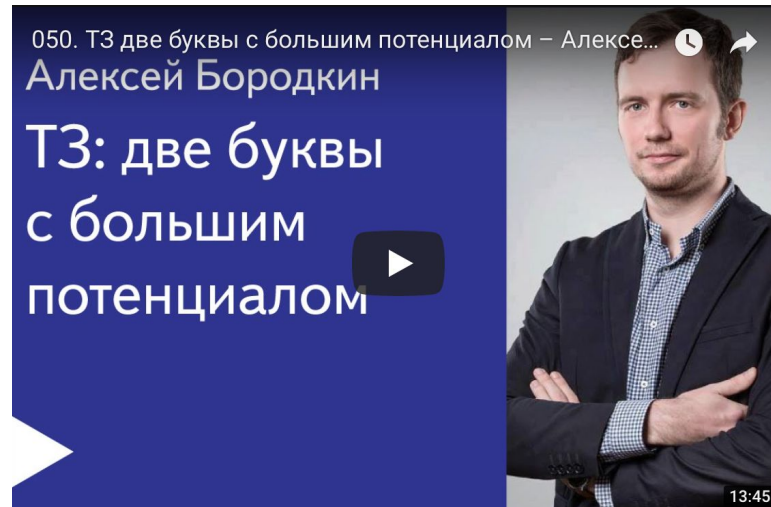
- ТЗ по ГОСТ
 - Много ненужного материала, “воды”
 - “Адская” структура и дублирование
 - Нет фокуса на продукт
 - Бессмыслица
- Что делать?
 - Делать все правильно в своем формате
 - Аутсорсить конвертацию в ГОСТ формат

Лекция А. Бородкина

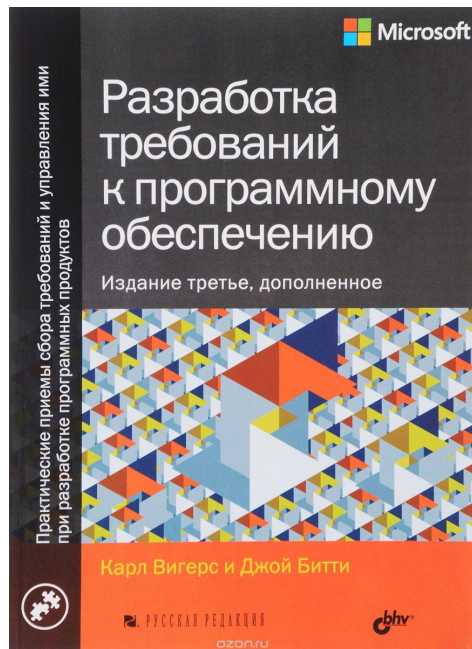
“Как правильно поставить ТЗ
на создание сайта”

Примеры документов Notamedia:

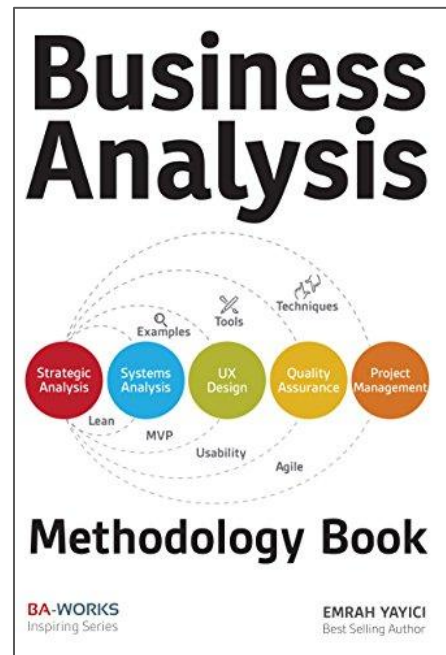
<http://nota.media/yandex/>



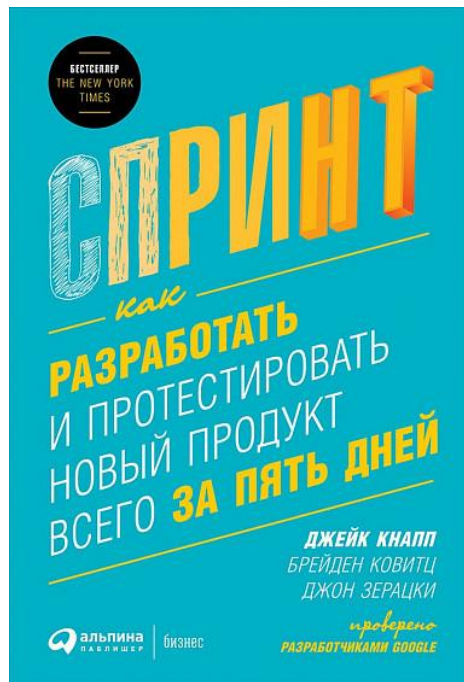
К. Вигерс, Дж. Битти
Разработка требований к ПО



Е. Yayici
Business Analysis Methodology Book



Дж. Кнапп, Б. Ковитц, Дж. Зерацки
“Спринт. Как разработать и протестировать
новый продукт всего за пять дней”



М. Хаммер, Дж. Чампи
Реинжиниринг корпорации



Интересно про бизнес-аналитиков

- А. Байкин. [Введение в профессию аналитика](#)
- Б. Шлаин. [Бизнес-аналитик – свой человек за линией фронта](#)
- [Ключевые качества бизнес-аналитика в ИТ](#)
- [Становление аналитика](#)
- [PM и BA — эти «тонкие» различия...](#)
- [Полезные навыки аналитиков. Как стать профессионалом](#)
- [Fundamental Skills for Business Analysts](#)

Полезные ссылки

- [10 Requirement Elicitation Techniques for Business Analysts](#)
- [Илья Корнипаев. Креативные техники сбора требований](#)
- [Александр Байкин. Введение в профессию аналитика](#)
- [Откровения продакт-менеджера](#)
- [Методы выявления вариантов использования](#)
- [Inbox Zero: Action-Based Email](#)
- [Inbox Zero: How to Become an Email Ninja](#)