

Анализ требований

Требование — это пригодное для использования представление потребности.

Требования сосредоточены на понимании того, какая ценность может быть получена в результате выполнения требования.

Требование — это:

1. условие или возможность, необходимые заинтересованной стороне для решения проблемы или достижения цели.
2. условие или возможность, которые должны быть выполнены или которыми должно обладать решение/компоненты решения для удовлетворения контракта, стандарта, спецификации или других официальных документов.
3. документированное представление условий или возможностей

Требование

Потребность — это проблема или возможность, подлежащая рассмотрению

Потребности (мне нужно...)	Требования
построить дом для семьи	<ul style="list-style-type: none">● В доме должно быть 3 спальни● Дом должен быть 2-х этажный● Полы должны быть деревянными
создать сайт продаж	<ul style="list-style-type: none">● На сайте должен быть каталог товаров● На сайте должна быть возможность добавлять товары в корзину● Клиенты должны иметь возможность оформить заказ

Этапы работы с требованиями



1. **Выявление:** определить и собрать все потребности и пожелания от пользователей, стейкхолдеров и других заинтересованных сторон.
2. **Анализ:** понять, какие требования можно реализовать, а какие требуют доработок, определить приоритетность и зависимости между требованиями.
3. **Документирование:** формализовать требования, чтобы они были понятны для всех членов команды.
4. **Утверждение:** получить согласие от всех стейкхолдеров на финальный список требований.

Этапы работы с требованиями

Управление требованиями

- Определение процесса управления изменениями.
- Анализ влияния изменений требований.
- Создание базовой версии и управление версиями требований.
- Ведение журнала изменений требований.
- Контроль за состоянием всех требований.
- Использование средств управления требованиями.
- Создание матрицы связей требований.

Классификация требований

Классификация требований по Вигнерсу и BABOK отражает разные подходы к структурированию и управлению требованиями, с акцентом на анализ и выполнение в рамках разработки систем.

Карл Вигерс — американский специалист в области разработки программного обеспечения и автор нескольких книг, посвящённых этой теме. Он является экспертом в области управления требованиями и процессами разработки ПО.

BABOK — это свод знаний по бизнес-анализу от Международного института по бизнес-анализу ИБА.



Бизнес-требования (business requirement) - отображение целей, задач и результатов, объясняющих, для чего было инициировано изменение и каким образом будет оцениваться успех его реализации

- Почему это нужно?
- Зачем этого хотят?
- Для чего что-то менять?

Требования заинтересованных сторон (stakeholder requirement) - описывают потребности заинтересованных сторон, которые необходимо удовлетворить, чтобы выполнить бизнес-требования

- Какие потребности есть у конкретных пользователей?
- Какую ценность они для себя ожидают?

Требования к решению (solution requirement) описывают возможности или качество решения, удовлетворяющие требованиям заинтересованных сторон

- Что должно быть сделано?
- Какими качествами должна обладать система?

Функциональные требования (functional requirement) - поведенческие возможности, которые должно предоставлять решение для удовлетворения потребностей заинтересованных сторон.

- Какие функции должна выполнять система?
- Что должны иметь возможность сделать пользователи системы?



Функциональным
требованием к пакету
для молока будет
«способность
удерживать жидкость
без утечек».

Нефункциональные требования (non-functional requirements) - описывают условия, при которых решение должно оставаться действенным, либо качества, которыми оно должно обладать

- Какими характеристиками должно обладать решение?



Нефункциональным
требованием к каске может
быть «не должна ломаться под
давлением менее 10 000
фунтов на квадратный дюйм».

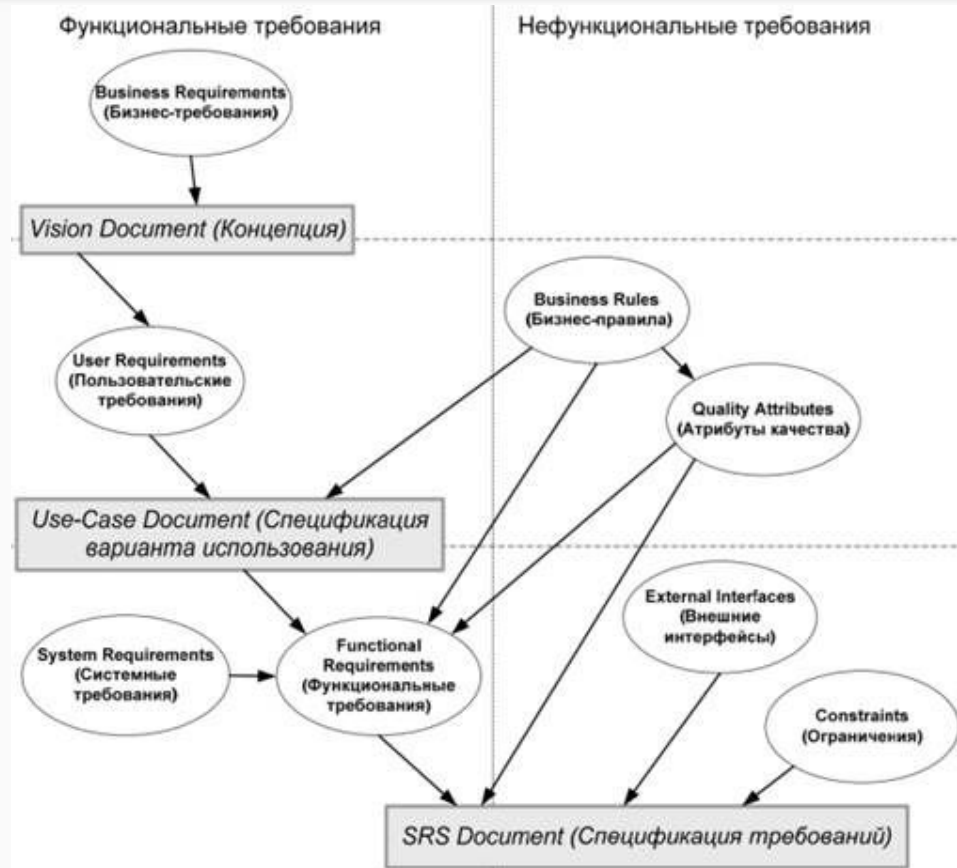
Переходные требования (transition requirement) описывают возможности решения и условия, каким оно должно соответствовать для перехода из текущего состояния в целевое.

- Что нужно сделать, чтобы внедрить новое решение?
- Что не нужно делать, чтобы решение было внедрено успешно?



Требование: Перенос данных из старой системы учета клиентов в новую CRM-систему.

Описание: При переходе на новую CRM-систему необходимо обеспечить корректную миграцию всех существующих данных о клиентах (включая контактную информацию, историю взаимодействий, финансовую информацию и статистику покупок) из старой системы. Миграция должна сохранить целостность данных и исключить их дублирование.



Бизнес-правила (business rules) включают корпоративные политики, правительственные постановления, промышленные стандарты и вычислительные алгоритмы.

Бизнес-правила не являются требованиями к ПО, потому что они находятся снаружи границ любой системы ПО.

Типы бизнес правил:

- **Факты**

Верное утверждение о бизнесе

“У каждого заказа есть стоимость доставки.”

- **Ограничения**

Определяет, какие операции НЕ могут или должны выполнять система и ее пользователи

“В переписке касающейся номера банковской карты клиента не может отображаться более 4-х цифр номера карты.”

“Все программные приложения должны соответствовать государственным правилам для использования лицами с ослабленным зрением.”

- **Вычисления**

Определяет вычисления с использованием математических формул и алгоритмов

*“Сумма позиции заказа рассчитывается как цена товара * на кол-во товара в позиции.” “Сумма НДС = сумма заказа * ставка НДС 20%”*

- **Термины**

Фиксируют значение тех или иных слов в проекте

- **Триггеры (активаторы)**

При определенных условиях приводит к выполнению каких-либо действий

«Если <некоторое условие верно или наступило определенное событие>, то <что-то произойдет>» — это ключ, который описывает активатор операции.

“После того, как клиент разместит книгу в корзине покупок, нужно ему показать связанные книги, которые другие покупатели также купили, когда они купили эту книгу.”

- **Производные (выводы)**

Устанавливает новые реалии на основе достоверности определенных условий

“Если продавец не может отправить заказанный товар в течение пяти дней с момента получения заказа, тогда товар считается возвращенным.”

Атрибуты качества (quality attributes) - дополнительное описание функций продукта, выраженное через описание его характеристик, важных для пользователей или разработчиков.

К таким характеристикам относятся:

- простота использования,
- производительность,
- целостность,
- доступность,
- устойчивость к сбоям
- масштабируемость

К типичным категориям нефункциональных требований относятся:

• Доступность	• Безопасность
• Масштабируемость	• Удобство использования
• Ремонтопригодность	• Сертификация
• Совместимость	• Соответствие
• Переносимость	• Локализация
• Эффективность работы	• Соглашение об уровне обслуживания
• Надежность	• Расширяемость

Внешние интерфейсы (external interfaces)

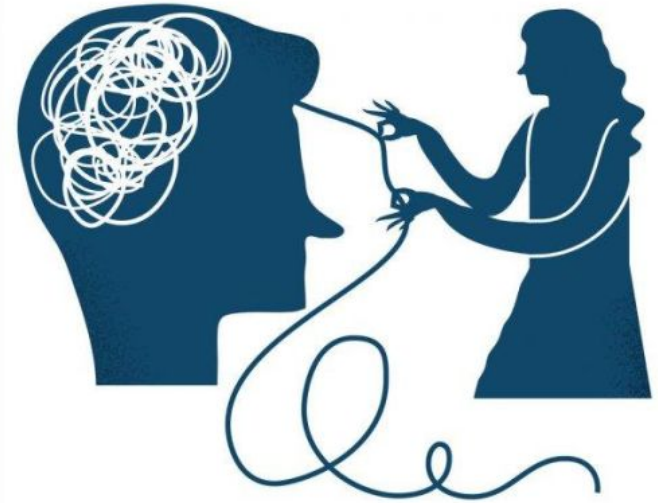
- Интерфейсы внешних систем
- Интерфейсы внешнего оборудования
- Интерфейсы пользователей

Ограничения (constraints) касаются выбора возможности разработки внешнего вида и структуры продукта.

Системные требования (system requirements) обозначают высокоуровневые требования к продукту, которые содержат многие подсистемы. Говоря о системе, мы подразумеваем программное обеспечение или подсистемы ПО и оборудования.

Методы сбора требований

- Интервью
- Анкетирование
- Мозговой штурм
- Проведение семинаров и воркшопов
- Работы с фокус-группами
- Анализ документации
- Изучение существующего решения
(анализ интерфейсов, reverse engineering и т.д.)
- Наблюдение
- Прототипирование



User Story

Пользовательская история (User Story) — это описание функциональной возможности ПО простыми, общими словами, составленное с точки зрения конечного пользователя. Она пишется с целью разъяснить, как именно функциональная возможность принесет пользу клиенту.

Как [категория пользователей]
Я хочу [определенная возможность/действие]
Чтобы [польза]

Критерии приемки:

- ...
- ...

Критерии приемки — это условия или пункты по которым мы будем принимать данную пользовательскую историю когда она будет реализована

- As a...
[Who? - Insert the User]
- I would like to...
[What? - Insert what they want to do]
- so that...
[Why? – Insert the Business Value]

User Story

Как пользователь интернет-магазина,
я хочу иметь возможность искать товары на сайте
чтобы подобрать и купить подходящий товар

Критерии приемки:

1. На вкладке со списком товаров система отображает поля поиска в верхней части экрана
2. При заполнении полей предпочтительными параметрами и нажатии кнопки “Применить”, система отображает элементы в разделе “Результаты поиска”, которые соответствуют выбранным критериям
3. Система отображает кол-во результатов поиска в разделе “Результаты поиска”

Follow the INVEST
guidelines for good
user stories!






Use Case

Вариант использования (Use case) — это перечень действий, сценарий по которому пользователь взаимодействует с приложением, программой для выполнения какого-либо действия для достижения конкретной цели.

- Имя сценария
- Цель
- Действующие лица
- Заинтересованные лица
- Предварительные условия
- Активаторы
- Порядок Событий
- Альтернативные пути и дополнения
- Исключения
- Результат
- Дополнения

Use Case

Имя сценария	UC-CR-06 — Регистрация на курс по системному анализу
Цель	прохождение курса по системному анализу
Краткое описание	вариант использования позволяет студенту записаться на курс по системному анализу с возможностью выбора даты начала курса
Действующие лица	Студент Система — web-приложение обучающего портала
Предварительные условия	Студент находится в каталоге с доступными курсами по системному анализу Установлен график проведения курсов Есть доступные курсы/курсы со свободными местами
Активаторы	Студент выбирает необходимый курс и нажимает кнопку «Регистрация»
Порядок событий	1. Система отображает окно в веб-приложении с графиком проведения курсов по выбранному курсу. 2. Студент выбирает предпочтительную дату начала курса 3. Система подтверждает, что курс доступен и есть свободное место на курсе 4. В открывшемся окне студент вводит личные данные для регистрации 5. Студент нажимает кнопку «Подтвердить» 6. Система регистрирует студента на курс по системному анализу на дату определенную на шаге 2. 7. Система отправляет студенту уведомление на почту об успешной регистрации на курс.
Альтернативный поток	4a Система заполняет личные данные студента автоматически, если студент выполнил вход в личный кабинет, и есть необходимые данные студента, необходимые для регистрации на курс. 5a. 5.a.1 Система выводит уведомление, если не все необходимые данные на странице регистрации на курс заполнены. 5.a.2 Студент вводит необходимые для регистрации личные данные и возвращается на шаг 5 основного сценария
Исключение	1a Система выводит уведомление «Нет доступных курсов», вариант использования завершается 6a. 6a.1 Система выводит уведомление студенту «Запись на курс закрыта» 6a.2 Система перенаправляет студента на страницу с каталогом доступных курсов по СА 7a. Если в процессе регистрации или отправки уведомления произошла системная ошибка, система отображает уведомление об ошибке пользователю, вариант использования завершается 8a. На любом шаге основного сценария, студент может нажать кнопку «Отмена» и система перенаправляет студента на страницу с каталогом доступных курсов по СА
Результат	Студент успешно зарегистрирован на курс по системному анализу на выбранную дату в обучающем портале Студент получил уведомление на почту об успешном зачислении на курс
Дополнение	Форма регистрации на курс представлена на странице  (здесь линк на страницу с описанием элементов формы  и вайрфреймом страницы) Регистрация на курс производится в системе согласно алгоритму  (здесь линк на страницу с описанием подробного алгоритма работы системы)

Use Case VS Use Story

User Story	Use Case
User Story короткие и описывают одно требование	более подробные и могут содержать несколько требований в рамках одного сценария использования
описывают требования с точки зрения пользователя	описывают сценарий использования системы.
сосредоточены на том, что пользователь хочет сделать	описывают, как система должна реагировать на действия пользователя.
могут быть легко изменены или переписаны в процессе разработки	более жесткие и могут быть более сложными для изменения
сосредоточена на одном пользователе или группе пользователей	может включать несколько акторов, включая внешние системы и другие роли.
обычно относится к одной итерации разработки	может охватывать несколько итераций или фаз проекта
более доступна для всех участников команды разработки, включая не технических специалистов	может быть сложным для понимания нетехническими участниками
фокусируется на коммерческой ценности продукта и потребностях пользователя	фокусируется на конкретных действиях и последствиях

Функциональное требование

Формула:

Условие + “система” + “должна выполнять”

Пример:

После ввода данных пользователем, система должна валидировать указанные значения согласно BR-17.

Пример:

Система отправляет электронное письмо с подтверждением при создании новой учетной записи пользователя.



Нефункциональное требование

Формула:

“система” + “должна работать так”

Пример:

Страницы сайта должны загружаться за 3 секунды при общем количестве одновременных пользователей <5 тысяч.

Пример:

Система должна обслуживать 20 миллионов пользователей без ухудшения производительности.

Такие требования определяют как должна работать система (насколько качественно), а не какие функции есть в этой системе

Сравнение подходов Вигнерса и BABOK

Подход Вигерса фокусируется на процессе сбора и формирования требований к программному обеспечению. Он показывает уровни требований от более общих к более частным, при этом важно повышать уровень детализации, приближаясь к началу разработки. Согласно Вигерсу, спецификация требований не должна содержать деталей дизайна или реализации, данных о планировании проекта или сведений о тестировании.

Подход BABOK ориентирован на задачи и методы бизнес-анализа. Руководство BABOK принято считать «библией» бизнес-аналитика, а любой набор практик, описанных в нём, должен быть адаптирован к конкретным условиям, в которых выполняется бизнес-анализ.

Управление требованиями

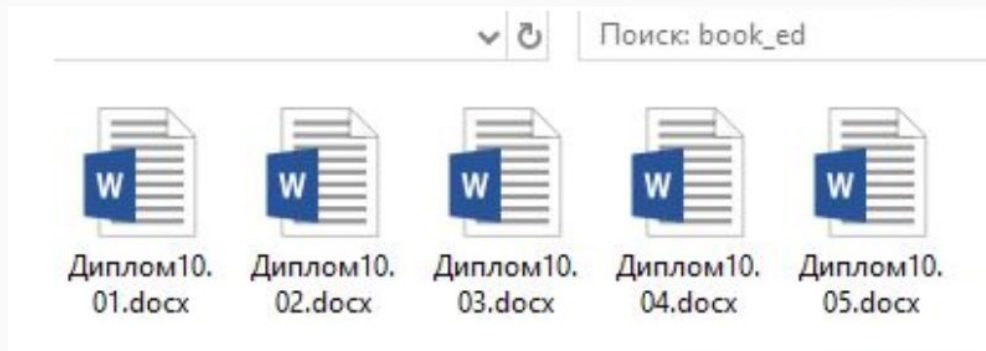


Управление требованиями

Управление версиями

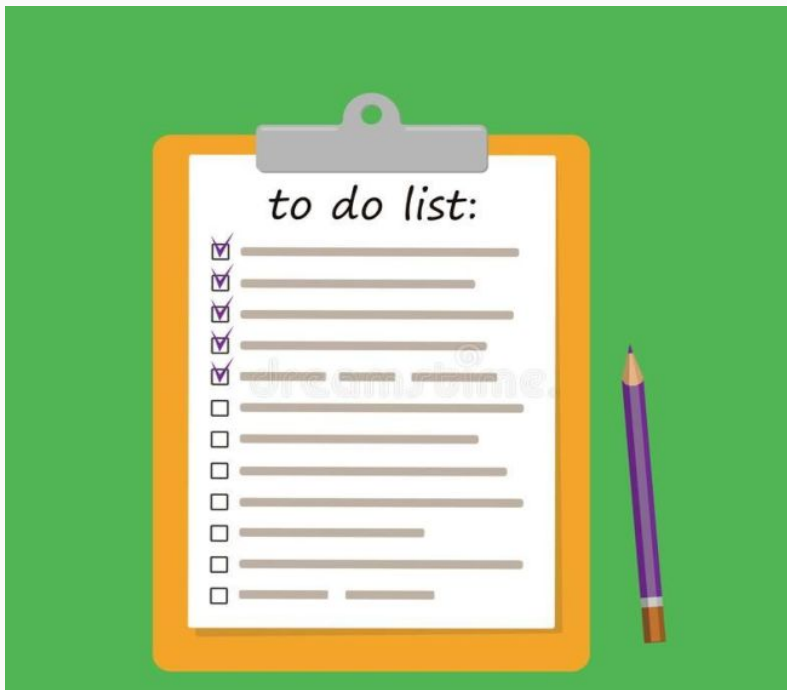
Вкратце:

Изменил требование – измени номер версии!



Управление изменениями

Требования меняются – это норма!



Управление изменениями:

1. Запрос на изменение
2. Анализ влияния изменений
3. Принятие решения
4. Внесение изменений в требования и артефакты
5. Согласование изменений
6. Обновление планов

Регламент внесения изменений

Регламент внесения изменений

(кто инициирует, в каком виде, кто принимает решение об изменении)
согласовываем при подписи (согласовании) документов, содержащих
требования!

