

# Функциональные спецификации и требования к контенту

Наумов Д.А., доц. каф. КТ

Компьютерная графика и проектирование графических интерфейсов, 2025

# Содержание лекции

1 Уровень набора возможностей

2 Анализ требований

## Определение набора возможностей

представляет ценность

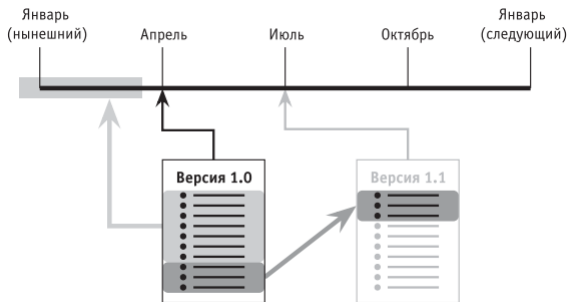
- как процесс – заставляет выявлять потенциальные противоречия и «шероховатости» конечного продукта на том этапе, когда сам результат существует лишь в вашей голове;
- как результат – дает точку отсчета для всей последующей работы над проектом и общий язык, на котором вы сможете обсуждать эту работу.

Имея четкое представление о том, чего хотим мы сами и чего хотят наши пользователи, мы можем понять, как достичь этих стратегических целей. Стратегия превратится в возможности системы, когда вы преобразуете потребности пользователей и цели сайта в конкретные требования к контенту и функциональности веб-сайта – к тому, что он предлагает пользователям.



## Необходимость описания набора возможностей:

- 1 вы будете знать, что именно вы создаете;
- 2 вы будете знать, что именно вы не создаете;
- 3 вы будете знать, что именно вы не создаете прямо сейчас;

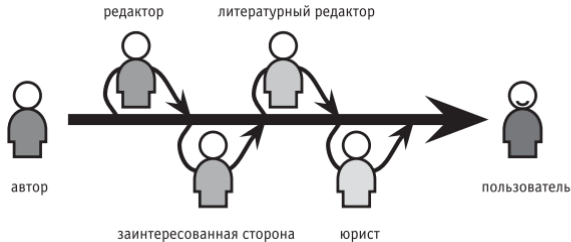


## Функциональность и контент

Набор возможностей определяется в документах, содержащих требования к функциональности, или в **функциональных спецификациях**.



При **разработке контента** определение требований часто не включается в формальную процедуру.



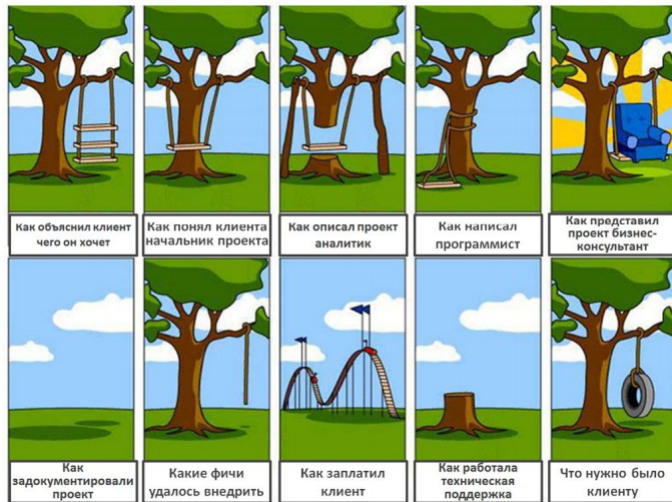
**Система управления контентом** способна автоматизировать рабочий процесс подготовки контента и донесения его до пользователя.

- Функциональность зависит от природы самого контента.
  - Нужно ли вам поддерживать несколько языков и различные форматы данных?
  - Должен ли каждый пресс-релиз визироваться юристом?
  - Будут ли элементы содержимого сайта автоматически переупорядочиваться в соответствии с предпочтениями каждого пользователя?
- Функциональные требования влияют на контент.
  - Потребуются ли инструкции на экране настроек?
  - Как насчет сообщений об ошибках?

# Требования к контенту

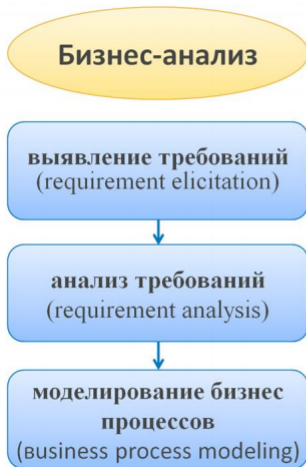
- количество слов текста (в статье);
- размеры изображений в пикселях;
- размеры отдельных скачиваемых файлов (PDF-документов, аудио- и видеофайлов);
- кто из сотрудников за какой элемент контента отвечает?
- как часто элемент контента он будет обновляться?

# Понятие анализа требований





# Понятие анализа требований



**Анализ требований** — это процесс сбора требований к ПО, их систематизации, документирования, анализа, выявления противоречий, неполноты, разрешения конфликтов в процессе разработки программного обеспечения.



# Структура анализа требований

## Сбор требований

общение с клиентами и пользователями, чтобы определить, каковы их требования



## Анализ требований

определение, являются ли собранные требования неясными, неполными, неоднозначными, или противоречащими, и затем решение этих проблем



## Документирование требований

Требования могут быть задокументированы в различных формах, таких как простое описание, сценарии использования, пользовательские истории, или спецификации процессов

# Виды требований по характеру

## ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

требования к поведению системы

**Функциональная целесообразность (functional appropriateness)** –  
реализованные функции соответствуют требуемым

**Функциональная полнота (functional completeness)** –  
полнота реализации требуемых функций

**Функциональная корректность (functional correctness)** –  
точность реализации требуемых функций

**Способность к интеграции (interoperability)** –  
наличие специальных входов и выходов для интеграции с другим ПО

**Безопасность (security)** –  
инкапсуляция состояний; все входы, выходы – ожидаемые

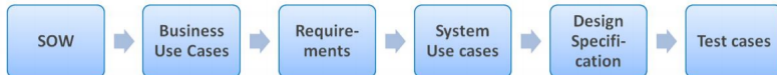
описание того,  
что система  
должна делать

# Виды требований по характеру

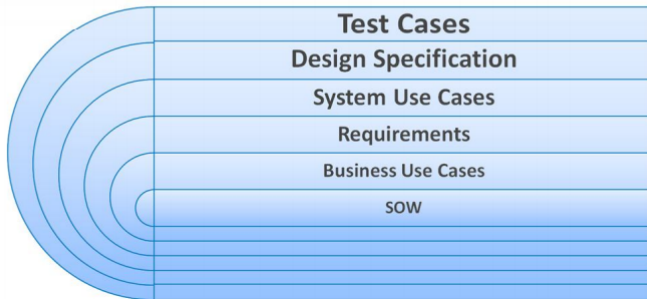


# Трассировка требований

Возможные уровни представлений требований по времени возникновения:



Возможные уровни представлений требований по зависимости (вложенности):



# Трассировка требований

- SOW – Statement of Work, требования высокого уровня, наименее конкретные, без четкой детализации.
- Business Use Cases – варианты использования (Use Case) на уровне бизнеса
- Requirements – список требований, разделенные на функциональные, нефункциональные и т.п. Структурированный документ.
- System Use Cases – варианты использования (Use Case) на уровне ПО Design Specification – проектная спецификация, техзадание (для разработчиков).
- Test Cases – варианты использования готового ПО, соответствия реального поведения и желаемого. Наиболее формализованные требования.

# Методы выявления требований

## Традиционные методы

Интервью

Анкетирование

Наблюдение

Изучение документов и программных систем

Мозговые штурмы, семинары

## Современные методы

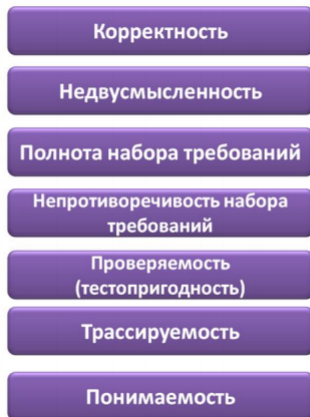
Прототипирование

Совместная разработка приложений (JAD-метод)

Быстрая разработка приложений (RAD-метод)



# Характеристики качественных требований





# Корректность

Насколько корректно наше требование? Действительно ли это то, что требуется от системы или кто-то допустил ошибку/опечатку в процессе написания требования?

**Пример:** после набора номера пользователь должен слышать короткие гудки (символизирующие о том, что идет дозвон)

**Описание:** просто перепутали слово. Гудки должны быть длинными. Как тестировать на корректность и находить такие ошибки:

- нужны хорошие доменные знания в области
- нужно смотреть на трассировку вверх и вниз, возможно обнаружится нестыковка и будет зацепка;
- процесс ревью также может помочь (желательно, чтобы это ревью проводил; тест-аналитик или тот, кто тоже имеет отношение к написанию требований).

# Недвусмысленность

Могут ли 2 различных человека понять требование по-разному?

**Пример:** Телефон должен работать в автономном режиме, когда он питается от батареек. В автономном режиме недоступны следующие функции: бла-бла-бла.

**Описание:** должны ли быть доступны функции «бла-бла-бла», если телефон с батарейками, но подключен к сети? Я могу подумать, что да, программист — что нет. Заказчик тоже может это интерпретировать как захочет

Как тестировать на корректность и находить такие ошибки:

- Проверять «ветвистость» требований
- Ревью, аналогично предыдущему пункту
- В идеале — стараться избегать ветвлений в требованиях. Если это невозможно — форматировать их в виде таблиц со всеми возможными вариантами.

## Полнота набора требований

Насколько полным является набор требований? Если есть секция в SRS, определяющая функциональность модуля, то вся ли функциональность этого модуля покрыта требованиями? Нет ли дыр?

**Пример:** Есть секция требований, определяющая работу со спец-кнопками, и в этих требованиях упущена из виду одна из спец-кнопок. Для этой кнопки просто нет требований.

Как тестировать на корректность и находить такие ошибки:

- нужно смотреть на трассировку требований вверх и вниз (на бизнес-требования и на низкоуровневые требования — дизайн, макеты, детальное описание реализации).
- Если в этих требованиях есть то, что упущено в SRS — такая ошибка сразу обнаружится.
- Если этого и там нет — то, возможно, этой функциональности и не должно быть? Или должна быть, но она упущена из виду во всех документах

# Непротиворечивость набора требований

Поиск требований, которые противоречат друг другу.

## Пример:

*Требование 1* (из раздела функциональности спец-кнопок): когда активен режим «Mute», телефон не должен издавать никаких звуков

*Требование 245* (из раздела интерфейса): когда пользователь снимает трубку, телефон должен издавать тоновые гудки

Как тестировать на корректность и находить такие ошибки:

- обращать внимания на общие формулировки в требованиях;
- делить на категории (например, выделить все требования, регламентирующие звуковые сигналы) и ревьюить их направленно на предмет противоречий;
- выделять все требования, трассирующиеся на одно верхнеуровневое требование и анализировать такие наборы.

## Проверяемость (тестопригодность)

Один из основных и самых важных критериев (для QA инженеров).  
Возможно ли проверить это требование и убедиться, что оно выполняется?

**Пример:** в случае возникновения критической ошибки телефон должен перезагрузиться

**Пример 2:** информация на дисплее телефона должна отображаться в понятном пользователю виде

Как тестировать на корректность и находить такие ошибки:

- Во время анализа требований задавать вопрос: «Как я буду это проверять?». Если однозначного ответа нет — значит нужно более детально анализировать, и, возможно, вносить правки в требование (уточнения, ограничения)
- Во время анализа требований выявлять общие формулировки, требующие перебора неопределенного числа вариантов для проверки выполнения требования.

# Трассируемость

**Трассируемость** — это связь с требованием выше и требованием ниже.

**Пример 1.** Бизнес-требование не имеет ни одного SRS требования. Соответственно, получается пробел в реализации (мы не сделаем того, что нужно бизнесу)

**Пример 2.** SRS требование описывает то, чего не было в бизнес-требованиях. Получается, мы делаем то, что не требовалось. Как тестировать на корректность и находить такие ошибки:

- если используется какая-то система менеджмента требований — то там, скорее всего, уже есть функциональность, позволяющая в автоматическом режиме следить за трассировкой
- если системы менеджмента требований нет, то остается "дедовский" способ — составлять матрицы трассируемости и отслеживать связи всех требований на всех уровнях.

# Понимаемость

Могут ли все участники процесса понять, что требуется от системы по описанию требования? Часто бывают ситуации, когда требование может быть понятно разработчику, но не понятно тестировщику (или наоборот). При этом требование может вполне соответствовать остальным критериям.

**Пример:** передатчик телефона должен использовать амплитудно-фазовую модуляцию с несущими от 2 до 7,5 МГц с шагом 500 кГц

Как тестировать на корректность и находить такие ошибки:

- стараться представлять себя на месте заказчика/аналитика/простого пользователя и пытаться представить, будет ли понятно это требование;
- обращать внимание на двойственные термины (особенно, аббревиатуры), которые могут интерпретироваться по-разному различными людьми.

# Критерии формулировки требований

**S****Specific** (конкретная)**M****Measurable** (измеримая)**A****Achievable** (достижимая)**R****Relevant** (сопоставимая)**T****Timebound** (определена во времени)

Достижение выполнения требований зависит от их формулировки, и первый шаг к успеху дела — правильно сформированные требования.