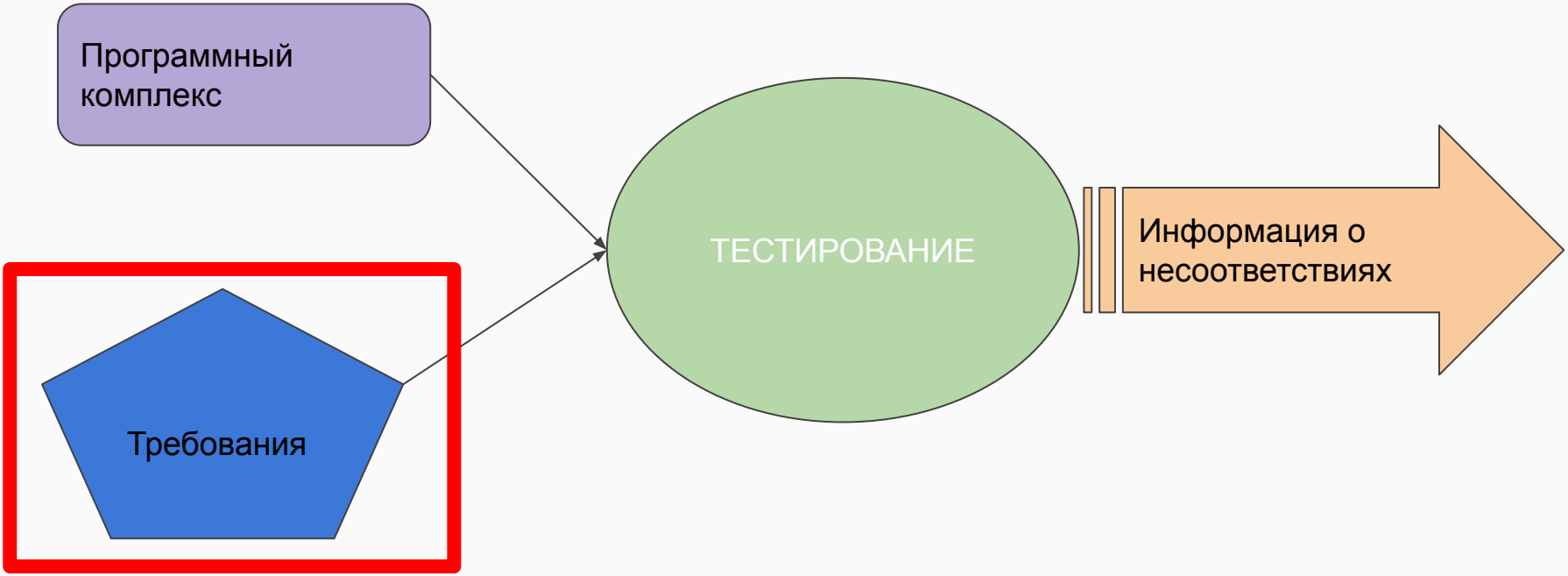
Manual QA course

Lecture 2. Требования к программному обеспечению и их анализ

Дорофеев Максим

hillel
компьютерная
школа

Схема процесса тестирования



Требования к программному обеспечению

- Некое **свойство** программного обеспечения, необходимое пользователю, для решения **проблемы** при достижении поставленной цели.

- Некое **свойство** программного обеспечения, которым должна обладать система или ее компонент, чтобы удовлетворить **требования** контракта, стандарта, спецификации либо иной формальной документации.

Требования к программному обеспечению

Требования бывают:

- **Прямыми**(Формализованными в технической документации, спецификациях, User Story)

- **Косвенными**(Проистекающими из прямых, либо являющиеся негласным стандартом для данной продукции или основывающиеся на опыте и здравом смысле использования продукта или продуктов подобных ему)

Виды требований к ПО по уровням

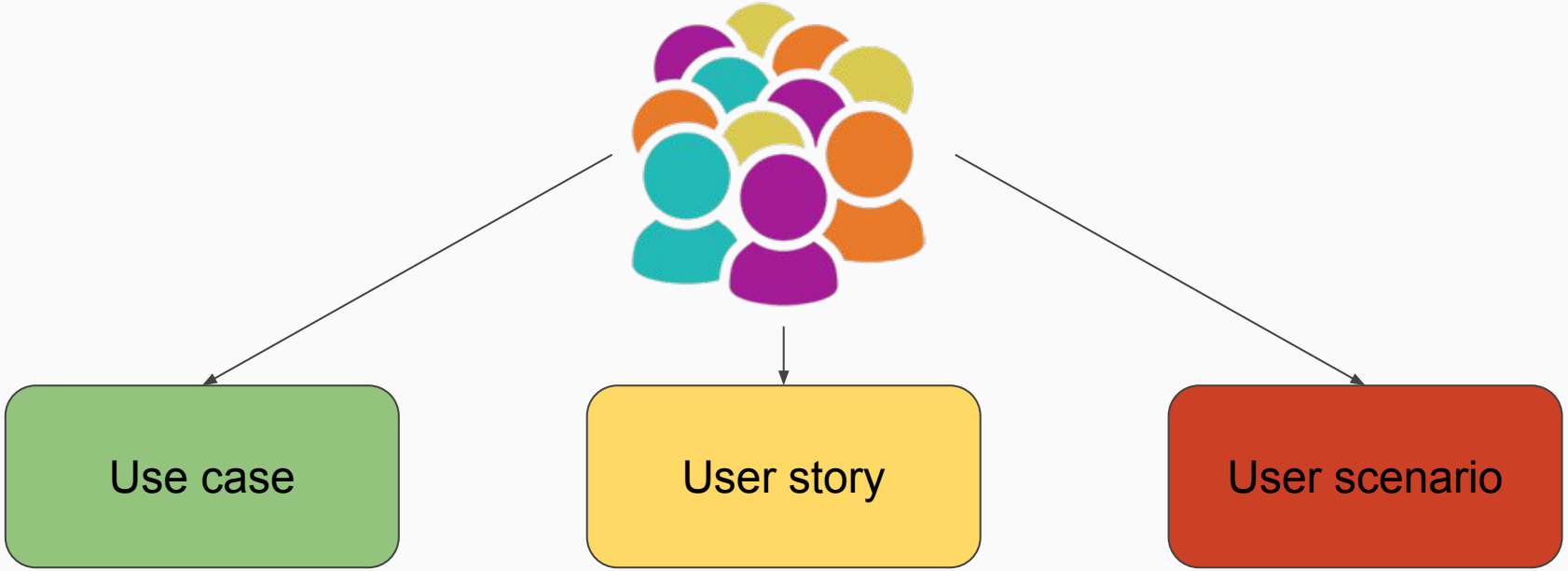
Требования бизнеса:

1. Высокоуровневые цели организации или заказчика (Контекст)
2. Цели, создания системы и критерии их достижения.
3. Ключевые требования к решению и их приоритеты.
4. Список **стейкхолдеров** (Лица заинтересованные в системе)
5. Ограничения на решения

Требования бизнеса: Приложения

1. Перечень бизнес – процессов.
2. Бизнес – правила.
3. Концептуальная модель предметной области

Пользовательские требования



Пользовательские требования. User Scenario

Терминал удостоверяется, что пополнение возможно, и запрашивает у **Пользователя** номер телефона и, если нужно, код оператора. **Пользователь** сообщает **Терминалу** запрошенные данные. **Терминал** удостоверяется, что данные введены корректно.

Пользовательские требования. User Story

Пользовательские истории — Способ описания требований, к разрабатываемой системе, сформулированный, как одно или более предложений на повседневном или деловом языке.

Цель **пользовательских историй** состоит в том, чтобы быть в состоянии оперативно и без накладных затрат реагировать на быстро изменяющиеся требования реального мира

Пользовательские требования. User Story

Типы:
Как <Роль/Персона пользователя> я <Хочу что – то получить>, <С такой – то целью>

Как <Пользователь>, я <Хочу управлять рекламными объявлениями>, <Чтобы удалять устаревшие или ошибочные объявления>

Пользовательские требования. Use Case

Use Case - Описание поведения системы, когда она взаимодействует с кем – то (или чем - то) из внешней среды. Система может отвечать на внешние запросы или сама выступать инициатором взаимодействия

Пользовательские требования. Use Case

1. Пользователь захотел разместить объявление
2. Пользователь зашел в систему
3. Пользователь авторизовался в системе
4. Пользователь создал объявление
5. Система отобразила сообщение об успешном создании объявления

Use Case для руководителя проекта

Обычно не содержит деталей реализации и пишется на языке целей пользователей. Поэтому **Use Case** удобно согласовывать с заказчиком как «**Единицу поставки**»

Use Case для разработчика

Когда он видит не отдельное «система должна…», а **контекст использования** той или иной функции. Какие функции будут выполнены, прежде чем будет вызвана эта? В каком виде и почему будут введены данные? Можно ли менять этот реализованный класс или это отразится на согласованных сценариях работы пользователя с системой?

Это **понимание** позволит разработчику лучше планировать работу над реализацией отдельных объектов и функций, а также снимет часть вопросов о используемых форматах данных.

Use Case для тестировщика

Use Case являются отличной базой для формирования тестовых сценариев — **Test Case**, — так как они описывают в каком контексте должно производиться каждое действие пользователя. **Use Case**, по умолчанию, являются тестируемыми требованиями так как в них всегда указана цель, которой нужно достигнуть и какие шаги надо для этого воспроизвести.

Use Case: Ограничения

Use Case не обеспечивают полноту всех функциональных требований, если в систему должна быть заложена сложная **бизнес-логика**.

Сценарии использования плохо подходят для документирования требований **не основанных** на взаимодействии с системой (таких как **алгоритм** или **математические требования**) или **нефункциональных требований** (такие как платформа, производительность, синхронизация, безопасность).

Следование шаблонам не гарантирует качество сценариев. **Качество** зависит только от навыков создателя сценария.

Use Case: Преимущества описания

- Дают представление о поведении системы.

- Понятны заказчика и разработчикам

- Позволяют описать множество альтернатив (Исключений)

- Список Use Case – перечень функциональности системы

- Для поддержки системы. Чтоб выявить ошибку, разобраться, на каком шаге что пошло не так.

Use Case: Рекомендации

- Основной сценарий не больше 3 – 9 шагов.

- Не включать элементы дизайна.

- Использование одного уровня детализации на всех шагах.

- Не использовать «Если»

Функциональные требования. Спецификация системы

Определяют характеристики ПО (**Функциональность**), которые разработчики должны построить, чтобы пользователи смогли выполнить свои задачи в рамках бизнес-требований.

Виды требований к ПО по характеру. Функциональные

Виды требований к ПО по характеру. Нефункциональные

- Ограничения
- Бизнес - правила
- Внешние интерфейсы
- Предложения по реализации
- Предложения по тестированию ПО
- Юридические требования
- Требования к безопасности

Источники требований

- Федеральное и муниципальное отраслевое законодательство (Конституция, законы, распоряжения)
- Нормативное обеспечение организации(Регламенты, положения, уставы, приказы)
- Текущая организация объекта автоматизации
- Модели деятельности(Диаграммы бизнес - процессов)
- Представления и ожидания потребителей и пользователей системы
- Опыт использования аналогичного ПО
- Конкурирующие программные продукты

Методы определения требований

- Анкетирование
- Мозговой штурм
- Наблюдение за производственной деятельностью
- Анализ нормативной документации
- Анализ моделей деятельности
- Анализ конкурентных продуктов
- Анализ статистики использования предыдущих версий системы

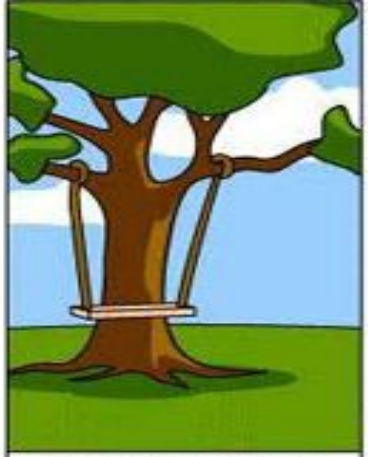
Качество требований

- Единичность
- Завершённость
- Последовательность
- Атомарность
- Отслеживаемость
- Актуальность
- Выполнимость
- Недвусмысленность
- Обязательность
- Проверяемость

Проверка требований

- **Тестирование**
- Анализ
- Осмотр
- Демонстрация

Как объяснил клиент чего он хочет

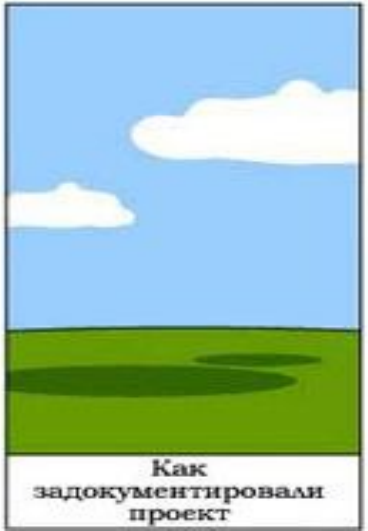


Как понял клиента начальник проекта

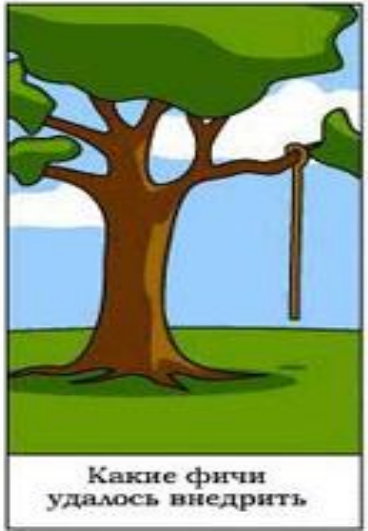
Как описал проект аналитик

Как написал программист

Как представил проект бизнес-консультант

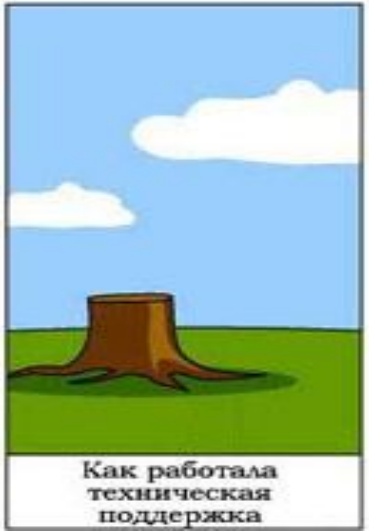


Как задокументировали проект



Какие фичи удалось внедрить

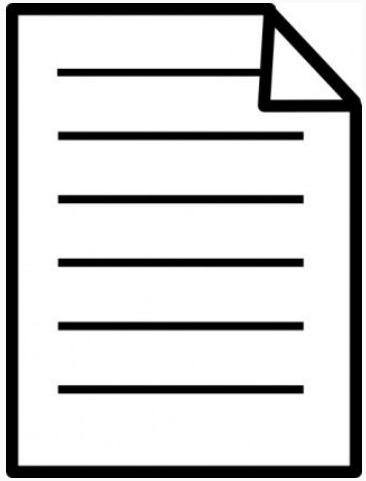
Как заплатил клиент



Как работала техническая поддержка

Что было нужно клиенту

Текстовая форма представления требований

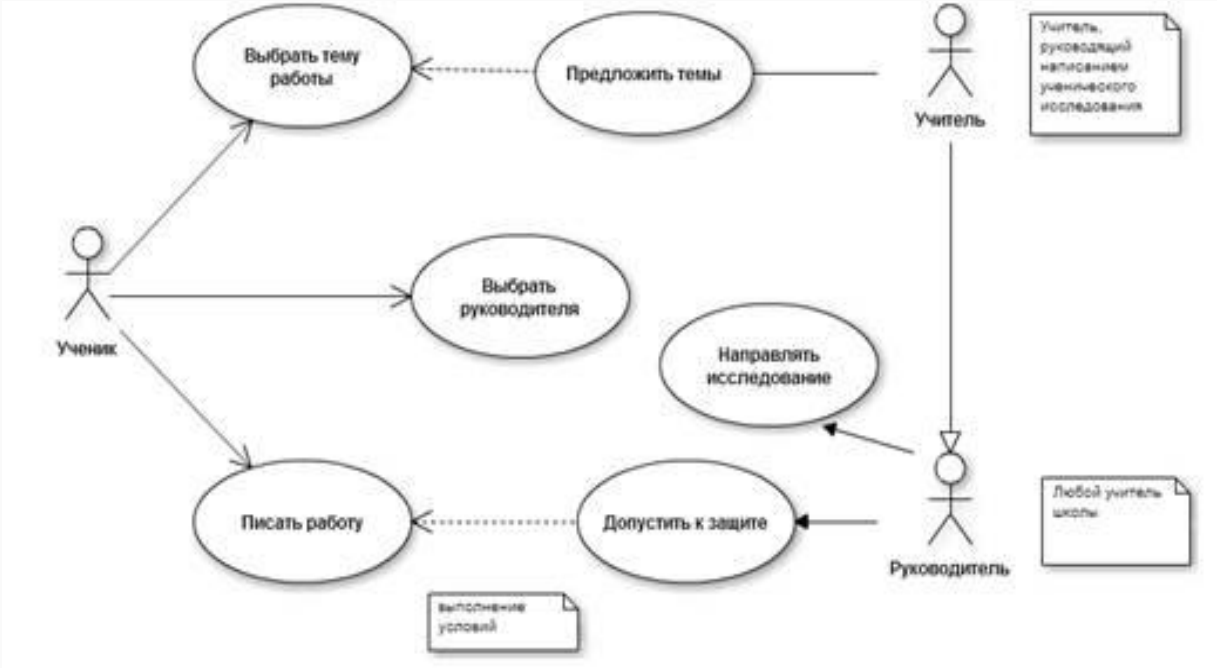


- Требования бизнеса
- User Stories
- Спецификация системы

Графическая форма представления требований

- ER (IDEF1FX), IDEF0, IDEF3
- DFD
- UML
- SysML

UML: пример


Выбрать тему работы
Предложить темы
Учитель
Учитель, руководящий написанием ученического исследования
Ученик
Выбрать руководителя
Направлять исследование
Писать работу
Допустить к защите
Руководитель
Любой учитель школы
выполнение условий

Вопросы и ответы

Ссылки

https://www.scrumalliance.org/community/articles/2013/september/agile-user-stories
https://habrahabr.ru/company/simbirsoft/blog/307844/
http://www.newlinestudio.ru/ArticleTrebovaniaPO.php
http://2tickets2dublin.com/how-to-write-good-user-stories-part-1/
Требования к ПО ВИКИ
http://www.dpgrup.ru/software-requirements.htm
http://www.comptechdoc.org/independent/programming/programming-standards/software-requirements-definition.html