



# \* Повторение урока №4

- Что такое тест кейс?
- Из чего он состоит?
- Что такое чек лист?
- Разница между тест кейсом и чек листом?
- Что такое тест план?
- Какие виды тест плана? Разница?
- Виды тест метрик?
- Тестирование? QA?
- Виды тестирования?

# \* Верификация и валидация. Что это и в чем разница?

Верификация (verification) - это процесс оценки системы или её компонентов с целью определения того, удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа. То есть, выполняются ли задачи, цели и сроки по разработке продукта.

Валидация (validation) - это определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе.

№	Верификация	Валидация
1.	<i>Делаем ли мы продукт правильно?</i>	<i>Делаем ли мы правильный продукт?</i>
2.	Реализована ли <i>вся</i> функциональность?	<i>Правильно ли</i> реализована функциональность?
3.	Верификация происходит раньше и включает проверку правильности написания документации, кода и т.д.	Валидация происходит после верификации и, как правило, отвечает за оценку продукта в целом.
4.	Производится разработчиками	Производится тестировщиками
5.	Включает статический анализ – инспектирование кода, сравнение требований и т.п.	Включает динамический анализ – выполнение программы для сравнения ее реальной работы с установленными требованиями
6.	Основывается на объективной оценке соответствия реализованных функций	Субъективный процесс, включающий личную оценку качества работы ПО

# \* Тестирование требований

Требования к требованиям:

Корректность

Недвусмысленность

Полнота набора требований

Непротиворечивость набора требований

Проверяемость (тестопригодность)

Трассируемость

Понимаемость

# \* Тестирование требований

## Корректность

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

## Недвусмысленность

Могут ли 2 различных человека понять требование по-разному?

## 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?

# \* Тестирование требований

Непротиворечивость набора требований

Поиск требований, которые противоречат друг другу. Это может быть очевидным кейсом, когда 2 требования явно говорят противоположные вещи, но может быть и скрытым, где противоречивость не очевидна на первый взгляд.

Проверяемость

(тестопригодность)

Для тестировщиков это — один из основных и самых важных критериев. Возможно ли проверить это требование и убедиться, что оно выполняется?

# \* Тестирование требований

Трассируемость

Любое требование проходит путь от бизнес-идеи до деталей реализации.

Это может быть 3 уровня требований (product requirements, software requirements, detailed design document), может быть и больше.

Трассируемость — это связь с требованием выше и требованием ниже.

Например, есть бизнес-требование о том, что должна быть возможность отключать звук. Оно может распадаться на много требований, описывающих функциональность режима Mute. Далее, это может быть еще детальнее расписано в design-e, где будет указано, как именно это реализовать. Связь между этими всеми требованиями — и есть трассировка.



# \* Тестирование требований

Как тестировать на понимаемость :

Стараться представлять себя на месте заказчика/аналитика/простого пользователя и пытаться представить, будет ли понятно это требование. Если нет — возможно, его нужно как-то переформулировать или упростить (вынося технические детали на другой уровень)



# \* Практическое задание:

- Знакомство с зефиром
- Создать набор тест кейсов для тестирования  
<http://test.trinetix.net/task/>
- Создать СВОЙ тест цикл и добавить туда все созданные СВОИ тест кейсы
- Провести тестирование по своему тест циклу
- Зарепортировать найденные по тест кейсам баг репорты в джиру























