



Лекция 3

Testing types



Agenda

- Разбор Д/З
- Testing types

HW:

- 1) Three amigos session: what is it ?
- 2) TDD and BDD. Can these things be used together?
- 3) Gherkin language: what is it ?

Three amigos session

Meeting overview:

- The business stakeholder provides a high-level perspective to each user story and its business value.
- The member of the quality assurance team discusses the test cases already created for these business requirements and provides insights around acceptance criteria.
- The developer discusses the architecture, the low-level design with the team and provides a big picture of dependencies, tech requirements, and assumptions.

During the meeting:

- The business representative should present the story.
- QA members should present the test cases.
- User stories are enhanced through the conversation, the same as the test cases.
- The team should validate dependencies and prerequisites.
- Assumptions should not exist. If there are doubts they should be clarified before the estimation session.

TDD (test-driven development) подход, основные принципы (red-green refactoring):

- сначала пишутся тесты (пишет девелопер), а потом код
- тесты всегда в статусе failed сначала, потому что код ещё не написан (red)
- пишется код и заново прогоняются тесты
- если все passed, то проводим рефакторинг кода (green)
- фокусировка идет не на том, как написать сразу качественный код, а сначала на том, как написать код, который будет соответствовать требованиям
- подход может быть использован не только в написании юнит тестов, но и для остальных уровней (интеграционные, UI)

NB! Можно писать только unit tests без использования TDD подхода (то есть пишем тесты, после написания кода)

BDD: behavior-driven development (вырос из TDD) подход, основные принципы:

- написание тестовых сценариев, которыми в последующем будет пользоваться вся команда (девелоперы для написания кода(тестов), тестеры для тестирования)
- BDD не заменяет обычный agile процесс, который существует на проекте
- обычно используется Gherkin

Gherkin — один из самых популярных domain-specific languages для написания use cases для ПО простым языком. Он позволяет разработчикам, тестировщикам и представителям бизнеса, участвующим в процессе, четко понимать требования проекта. Более того, Gherkin действует как документация и скелет автоматизированных тестов.

Официальный стандарт языка Gherkin поддерживается Cucumber — инструментом тестирования программного обеспечения с открытым исходным кодом, поддерживающим BDD.

Чтобы написать сценарии на Gherkin, необходимо знать основные используемые ключевые слова:

- Feature
- Scenario
- Given
- When
- Then
- And
- But

Example.

Feature: As a User I want to log in into the System

Scenario 1: A User uses valid credentials

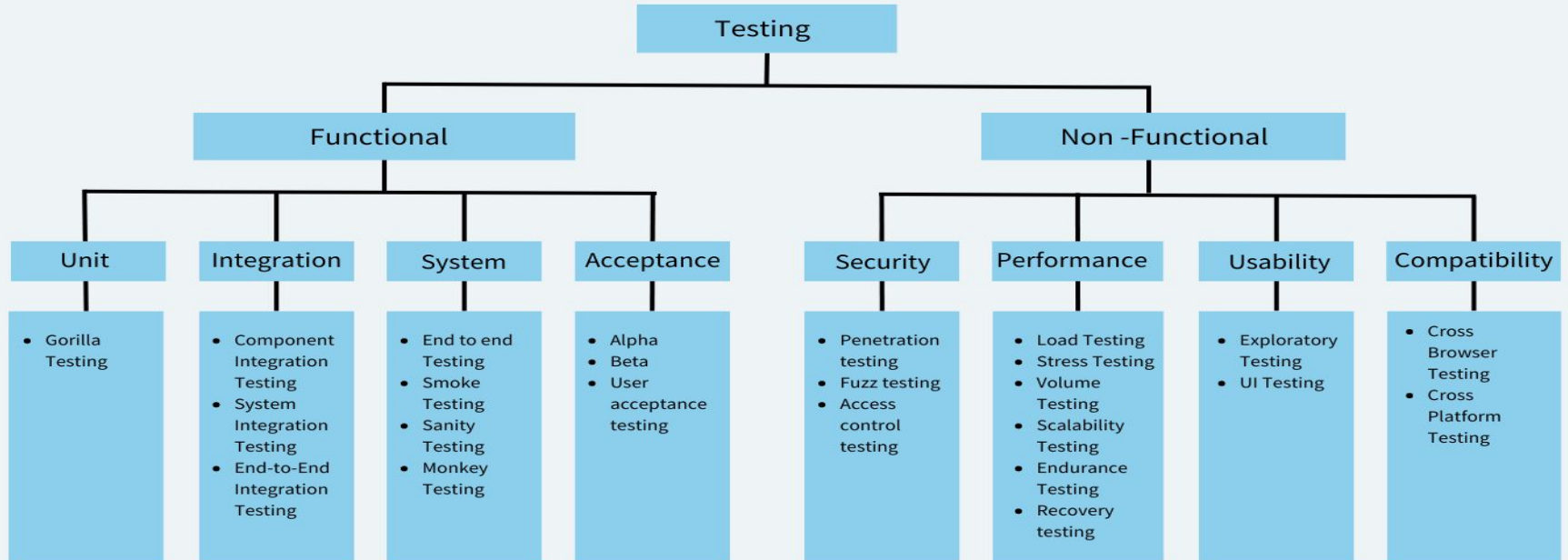
Given: User is on the login page of the System

When: User enter a valid username "Username"

And: User enter a valid password "Password"

Then: User is successfully logged in into the System

Testing types



Static testing is an approach to testing the software application without executing the actual code. This is performed at the early stage of development to identify the issues in the project documents in multiple ways, namely reviews, walkthroughs, and inspections

Dynamic testing is the type of testing that validates the functionality of an application when the code is executed / by executing the code. In simple terms dynamic testing is performed by actually using the application and seeing if a functionality works the way it is expected to.

Black-box testing - It is a type of software testing, in which the tester is not required to have any knowledge of the internal architecture or implementation of the system, for carrying out testing. It is performed, considering the software application as black-box only wherein we pass input to the application and verify the actual result with the expected result without considering how the data is getting processed internally.

White box testing - It is a type of software testing, in which the tester needs to have access to and knowledge of the internal architecture of the application. The tester analyses the architecture as well as the source code.

Вопрос: может ли мы сказать, что static testing can be a part of white box testing?

Top-down Integration testing

In top-down integration, testing starts from top modules to lower-level modules. At times, the lower-level modules are not created by the time top-level modules are tested. In those cases, 'Stubs' are created. These stubs are nothing but dummy modules that simulate the functionality of the lower-level modules.

Bottom-up Integration testing

In bottom-up integration, testing starts from lower-level modules to higher-level modules up in the hierarchy. Similar to top-down integration, the higher-level modules might not have got created by the time lower-level modules are tested. In those cases, 'Drivers' are used. These drivers are dummy modules that simulate the functionality of higher-level modules.

Load testing - refers to the practice of modeling the expected usage of a software program by simulating multiple users accessing the program concurrently

Stress testing - is a technique to test a system's performance under extreme conditions.

The key difference is the goal of each: Load tests help you understand how a system behaves under an expected load. Stress tests help you understand the upper limits of a system's capacity using a load beyond the expected maximum.

HW (это практика поможет потом в написании тестовой стратегии):

- 1) Изучить не рассмотренные типы тестирования и подумать, на каком проекте они могут пригодиться (тип тестирования, что это такое, где может применяться и как, куда можно внедрить)
- 2) Можно подумать и над рассмотренными типами (тип тестирования, что это такое, где может применяться и как, куда можно внедрить)

The End for today

