Pentalog

Software testing

PentaStagiu Remote Brașov

November, 2019 – March, 2020

DATA ANALYSIS
UX/UI
FRONT-END
TECHNOLOGY
NEAR/OFFSHORE
AGILITY
BACK-END
SPRINT
KANBAN
CAMPAIGNS
GROWTH HACKING
SCRUM
BACKLOG
DEVOPS
DESIGN
SEO
CONTINUOUS INTEGRATION
MOBILE
QA
AUTOMATION
RESPONSIVE
UNIT TESTING

Agenda

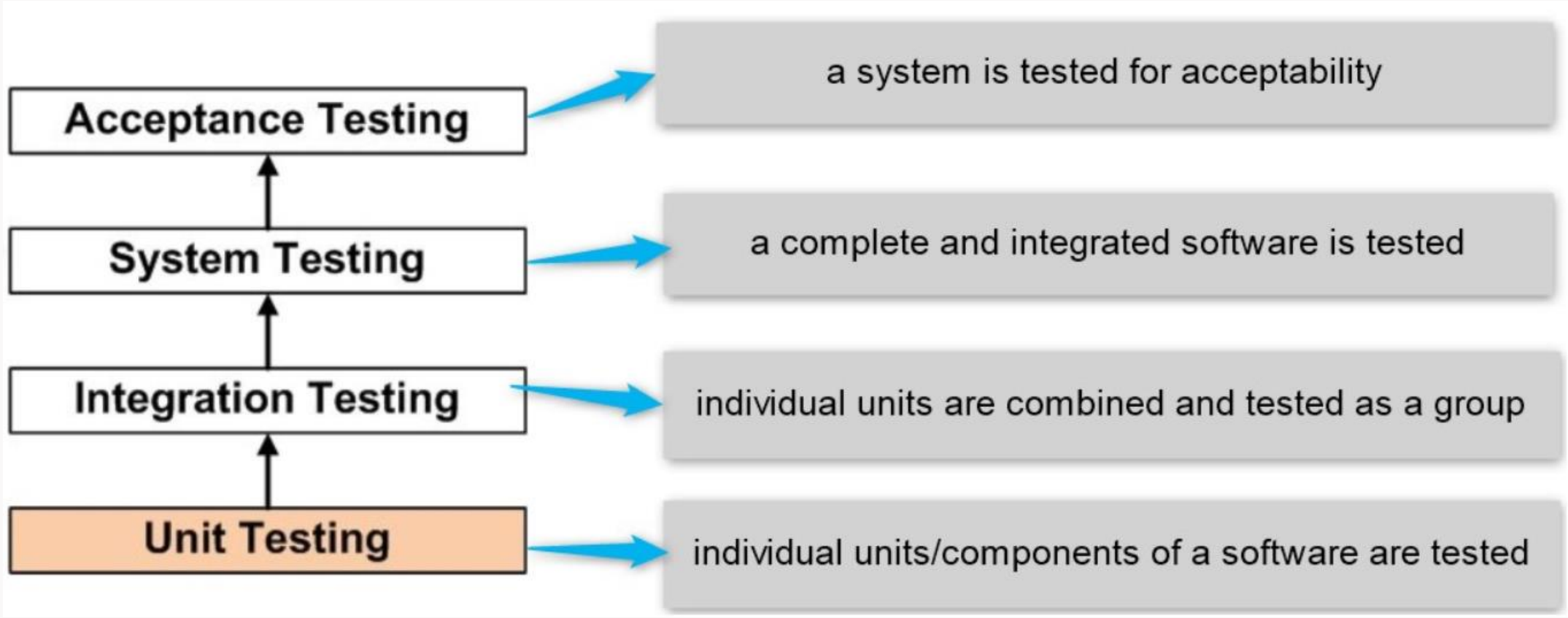
- What is Software Testing?
- Unit Testing
- Integration Testing
- Test Driven Development
- Nunit & Moq

What is Software Testing?

- Software testing is a process of executing a program or application with the intent of finding the software bugs

- It can also be stated as the **process of validating and verifying** that a software program or application or product:
  - Meets the business and technical requirements that guided it’s design and development
  - Works as expected
  - Can be implemented with the same characteristic.

What is Software Testing?



Unit testing

- is a level of software testing where individual units/ components of a software are tested

- The purpose is to validate that each unit of the software performs as designed

- A **unit:**
  - is the smallest testable part of any software
  - usually has one or a few inputs and usually a single output

- it is normally performed by software developers themselves or their peers. In rare cases, it may also be performed by independent software testers.

Unit testing benefits

- increases confidence in changing/maintaining code
- Codes are more reusable (In order to make unit testing possible, codes need to be modular)
- The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels

Unit testing tips

- Find a tool/framework for your language
- Do not create test cases for everything. Instead, focus on the tests that impact the behavior of the system
- Use test data that is close to that of production
- before fixing a defect, write a test that exposes the defect
- make sure you are using a version control system to keep track of your test scripts
- Perform unit tests continuously and frequently

Integration testing

- is a level of software testing where individual units are combined and tested as a group
- the purpose of this level of testing is to expose faults in the interaction between integrated units
- developers themselves or independent testers perform Integration Testing

Integration testing

- is a level of software testing where individual units are combined and tested as a group
- the purpose of this level of testing is to expose faults in the interaction between integrated units
- developers themselves or independent testers perform Integration Testing

Integration testing benefits

- it makes sure that **integrated** modules work properly as intended
- the tester can start testing once the modules to be tested are available
- it detects errors related to the interface between modules
- Helps modules interact with API’s and other third-party tools
- Typically covers a large volume of the system, so more efficient

Integration testing tips

- Ensure that you have a proper Detail Design document where interactions between each unit are clearly defined

- Make sure that each unit is unit tested before you start Integration Testing

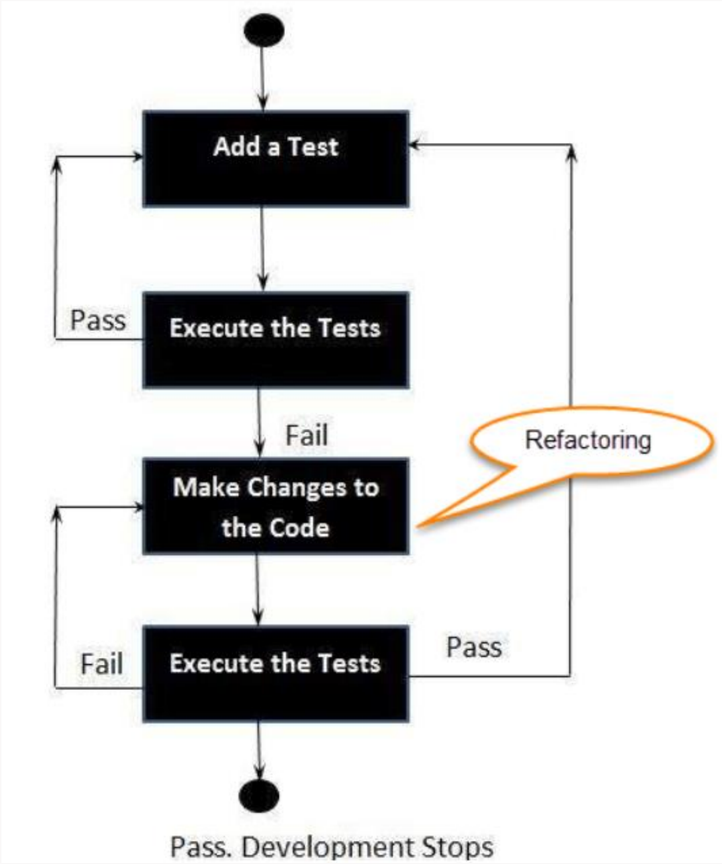
Test driven development

- Test-Driven Development starts with designing and developing tests for every small functionality of an application

- TDD instructs developers to write new code only if an automated test has failed

- Following steps define how to perform TDD test:

    1. Add a test.
    2. Run all tests and see if any new test fails.
    3. Write some code.
    4. Run tests and Refactor code.
    5. Repeat.



NUnit

- is a unit-testing framework for all .Net languages (open source)

- https://github.com/nunit/docs/wiki/NUnit-Documentation

Most used attributes:

- TestFixture - marks a class as a test fixture and may provide inline constructor arguments
- Setup - Indicates a method of a TestFixture called just before each test method
- TearDown - indicates a method of a TestFixture called just after each test method
- Test - marks a method of a TestFixture that represents a test
- TestCase - marks a method with parameters as a test and provides inline arguments
- Values - provides a set of inline values for a parameter of a test method

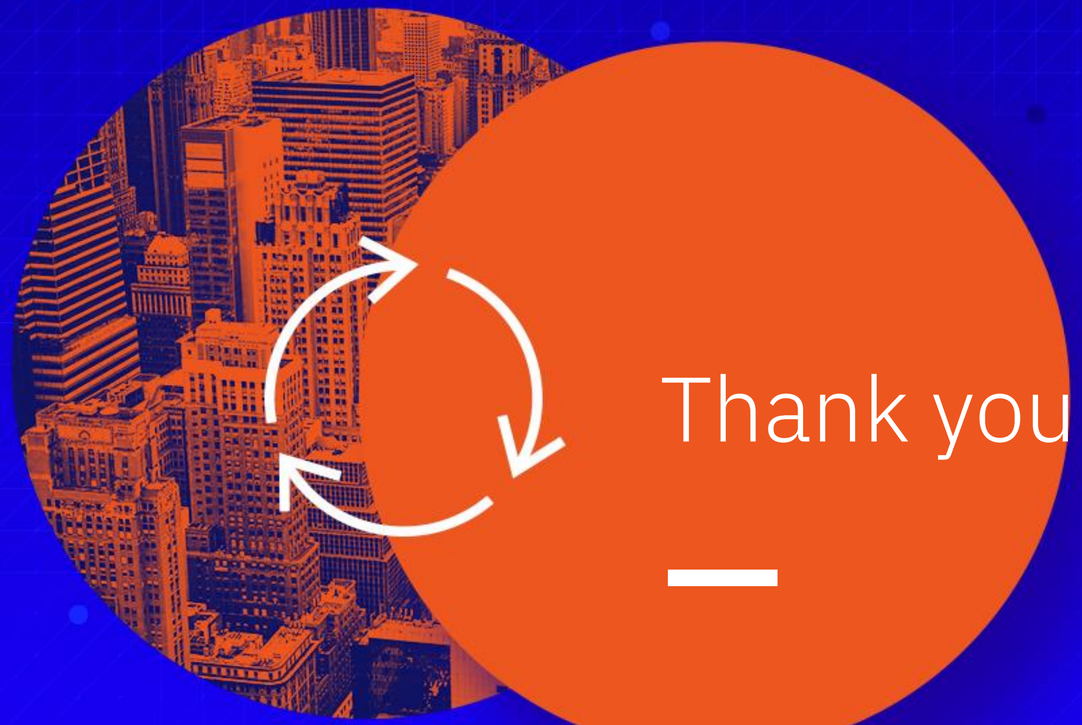
Moq

- The most popular and friendly mocking library for .NET
- **Mock** is a method/object that simulates the behavior of a real method/object in controlled ways
- https://github.com/Moq/moq4

Reference

- http://softwaretestingfundamentals.com/unit-testing/
- http://softwaretestingfundamentals.com/integration-testing/
- https://github.com/nunit/docs/wiki/NUnit-Documentation
- https://github.com/Moq/moq4

Pentalog



Thank you!

DATA ANALYSIS

UX/UI

FRONT-END

TECHNOLOGY

NEAR/OFFSHORE

AGILITY

BACK-END

SPRINT

KANBAN

CAMPAIGNS

GROWTH HACKING

SCRUM

BACKLOG

DEVOPS

DESIGN

SEO

CONTINUOUS INTEGRATION

MOBILE

QA

AUTOMATION

RESPONSIVE

UNIT TESTING