

Assignment 4 Levels of testing



Assignment Objectives:

- Generating test cases for different levels of testing.
- Levels of testing: unit, integration, functional, regression, acceptance.
- Use JUnit for implementing the test cases.
- Use Jenkins for Continuous Integration.
- Use Testlink for test case management.



Theoretical aspects

- Generating test cases for different levels of testing.
- Continuous Integration.
- References: Unit testing [Myers] – chapter 5; [Naik] – chapter 3
Integration testing - Techniques: [Naik] – chapter 7
Regression testing - [Young] –chapter 22
Function[al] testing [Myers] – chapter 6; [Naik] – chapter 9; [Young] –chapter 10 •
System testing [Myers] – chapter 6; [Naik] – chapter 8; [Young] –chapter 22 •
Acceptance testing [Naik] – chapter 14; [Young] –chapter 22

[Myers] Glenford J. Myers, *The Art of Software Testing*, John Wiley & Sons, Inc., 2004

[Naik] K. Naik, P. Tripathy, *Software testing and quality assurance. Theory and Practice*, A John Wiley & Sons, Inc., 2008

[Young] M. Pezzand, M. Young, *Software Testing and Analysis: Process, Principles and Techniques*, John Wiley & Sons, 2008

[TestLink1]: <http://www.softwaretestinghelp.com/testlink-tutorial-1/>

[TestLink tutorial]: <http://www.cs.ubbcluj.ro/~avescan/?q=node/189>



Assignment

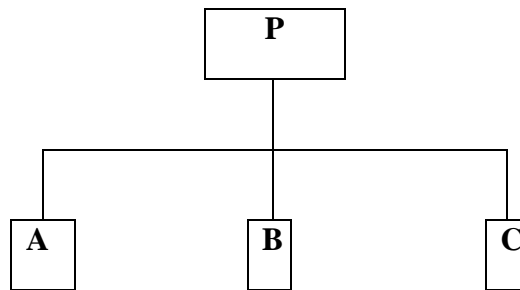
[Unit testing]

Create test cases using black-box testing and white-box testing for each project requirement a), b), c). The test cases were created in Lab 02 and Lab 03 (BBT for requirement a), WBT for requirements b)). For Lab 04 just test cases (BBT and WBT for c) and create the Jenkins Jobs and execute them (using Maven project).

[Integration testing]

The application has the following diagram dependency between modules, where:

1. **P** – the final application (with console interface);
2. **A** – the module for functionality a)
3. **B** – the module for functionality b)
4. **C** – the module for functionality c)



Use integration testing:

1. Big-Bang integration

- unit test A
- unit test B
- unit test C
- integration testing: all A, B, C.

Remark: You will create a test class having 4 test methods, one for each point above, 3 test cases calling one functionality and the 4th test case calling all 3 functionalities.

2. Incremental integration (top-down):

- unit test A.
- integration test B (A and B).
- integration test C (A and B and C).

Remark: You will create a test class having 3 test methods, one for each point above, 1 test case calling one functionality, 1 test case calling 2 functionalities (A and B) and 1 test case calling all 3 functionalities.

[Maven]

See Maven Tutorial

[Jenkins]

See Jenkins Tutorial

[TestLink]

a) Define the designed test cases for Integration testing (the ones that could be implemented and executed) in Testlink. See **Tutorial on Testlink**.

b) Define the requirements using Requirement Specification section. (if not created for Lab 02 and Lab 03)

c) Associate the test cases and the requirements to your Test plans.

Remarks

1. Use JUnit platform for testing - JUnit 3.x/4.x;



Turn in:

[Integration testing]

1) Source code:

- a) Implementation of the test cases for Integration testing in JUnit.
- b) The source code - tested and debugged (and retested).

[Jenkins]

The created Jobs in Jenkins for each type of testing:

- Unit testing: Jobs for BBT for Lab 02 and WBT for Lab 03, i.e. `scsIdJob_BBT`, `scsIdJob_WBT`.
- Integration testing: Job for Integrations, `scsIdJob_BigBang_IntegrationT`, `scsIdJob_Incremental_IntegrationT`.

[TestLink]

1) The Testlink - TestPlan/Test (the name of the test plan will be formed using your `scsId`, `scsId_TestPlan_BigBang_Integration` and `scsId_TestPlan_Incremental_Integration`). Cases must be presented/delivered in class.

2) The Testlink generated documentation from the Test Report menu.

For each type of testing [Unit testing], [Integration testing] generate the Test Report.

- In Testlink select Test Report menu
- Report format select MS Word
- Select the test plan (BBT, WBT, Integration)
- Click on Test Report. check all options, and click on the test plan.
- Save the document.

Assignment and Delivery date:



1. Assignment date: laboratory 4
2. Delivery date (first): laboratory 5
3. Delivery date (last): laboratory 6 (-1 point for each week late delivery)