# VVSS, Lab04: Niveluri de testare

## **Obiective**

- Testarea aplicaţiilor la diferite niveluri de testare, e.g., testare unitară, testare de integrare.
- Utilizarea tool-urilor folosite în procesul de testare (TestLink, Jenkins, JUnit, Maven, Git, Mockito).

## **Cerințe**

Să se realizeze următoarele task-uri:

Task, puncte	Descriere task					
7.	Identificați la nivelul arhitecturii clase particulare, codificate aici prin E, R și S care înplinesc următoarele					
	condiții: E reprezintă o entitate din domeniul problemei. R este un repository cu elemente de tip E. S					
	permite gestionarea repository-ului.					
[Unit	Realizaţi testarea în izolare (unit testing) pentru clasele E, R şi S, folosind framework-urile <b>JUnit</b> (vezi					
Testing.	Tutorial JUnit în Lab02) și Mockito (vezi Tutorial Mockito). Pentru R și S se vor descrie teste care					
JUnit.	folosesc <i>mock</i> sau <i>spy</i> , a alegere. Pentu fiecare clasă testată se va descrie câte o clasă de test separată					
Mockito]	cu minimum 2 teste.					
2 puncte	Nu se cere:					
	<ul> <li>elaborarea de fişiere similare cu Lab02_BBT_TCs_Form.xls şi Lab03_WBT_TCs_Form.xls;</li> </ul>					
	<ul> <li>crearea în TestLink a suite-lor de teste pentru testarea unitară modulelor E, R şi S.</li> </ul>					
[Integration						
Testing.	E, R şi S corespund claselor deja testate în izolare.					
Mockito]						
2 puncte	T T					
	S					
	R					
	Realizați testarea de integrare utilizând strategia de <b>integrare incrementală top-down</b> . Se vor evidenția					
	următoarele clase cu teste:					
	[task-ul anterior] Step 1. testare unitară pentru E, R şi S;					
	<b>Step 2.</b> integrare R (se testează S cu R; pentru E se folosesc obiecte <i>mock</i> );					
	Step 3. integrare E (se testează S + R cu E);					
	Pentru <b>Step 2.</b> şi <b>Step 3.</b> fiecare clasă de test va avea <b>minimum 2 teste</b> .					
[TestLink]	În cadrul proiectului <b>PrjAAA</b> , corespunzător userului <b>xyir1234</b> utilizat anterior pentru Lab03, se vor realiza următoarele task-uri:					
2 puncte	1.1. definiți planul de testare xyir1234_IntT_TP în cadrul proiectului PrjAAA (secțiunea <i>Test Plan</i> );					
	1.2. creați suita de teste <b>xyir1234_IntT</b> care va conține <b>3 cazuri de testare</b> , câte un caz de testare					
	pentru fiecare pas de aplicare a strategiei de integrare <b>Top down</b> (secţiunea <i>Test</i>					
	Specification);					
	1.3. asociați cazurile de testare create la planul xyir1234_IntT_TP;					
	1.4. asociați cazurile de testare create la cerințele create anterior, după caz, la xyir1234_F01 sau					
	xyir1234_F02;					
	1.5. generați documentația aferentă din (secțiunea Requirement Specification Document,					
	opţiunea <i>Show as <b>Pseudo MS Word</b></i> ).					
[Jenkins]	Se va crea, configura și executa <b>câte un job</b> (vezi <b>Tutorial Jenkins</b> ) pentru fiecare plan de testare creat					
2 puncte	anterior, i.e., xyir1234_BBT_TP, xyir1234_WBT_TP, xyir1234_IntT_TP. Job-urile create vor fi denumite					
	xyir1234Job_BBT, xyir1234Job_WBT şi xyir1234Job_IntT.					
	Observație:					
	•pentru fiecare test creat în TestLink se va verifica (1) dacă este un test executat automat și (2) dacă					
[TestLink+	are setat corect <i>numele clasei de test</i> şi <i>numele metodei de test</i> (vezi <b>Tutorial TestLink</b> , pagina 14).  Vizualizarea rezultatului execuției fiecărui job Jenkins în TestLink (vezi <b>Tutorial Jenkins</b> ). Pentru fiecare					
Jenkins]	test case executat este necesar să apară statusul <b>passed</b> .					
1 punct	test case executat este fiecesal sa apara statusui passeu.					
[Git]	Se va actualiza conţinutul repository-ului <b>Git</b> cu documentele elaborate în cadrul acestei teme:					
1 punct						
_ panet	<ul> <li>pachetul/ele cu teste implementate în Java pentru Lab04;</li> </ul>					
	<ul> <li>dacă este cazul, codul sursă modificat în urma depanării.</li> </ul>					
	adda este cazar, codar sarsa modificat in arma departarii.					

#### **IMPORTANT!!!**

Inainte de predarea temei de laborator, fiecare echipă va completa formularul de la acest link.

## Predarea temei de laborator

## [Unit Testing. JUnit. Mockito]

• implementarea claselor cu teste pentru clasele E, R şi S.

## [Integration Testing. Mockito]

• implementarea claselor cu teste folosind strategia de integrare **Top Down**.

#### [TestLink+Jenkins]

- Câte un job pentru fiecare dintre cele trei planuri de testare create în TestLink: xyir1234Job\_BBT, xyir1234Job\_WBT şi xyir1234Job\_IntT.
- Fiecare test din TestLink are statusul passed.

## Termene de predare

Săptămâna	Tema de laborator			Primul termen de predare	Ultimul termen de predare
S07	L04.	LO4. Niveluri de testare Test Jenkir	JUnit, TestLink,	S09	S11*°
S08			Jenkins, Git, Mockito	S10	S12*°

<sup>\*)</sup> Temele restante se vor putea preda în limita timpului disponibil.

<sup>°)</sup> Se pot preda cel mult două teme de laborator.