

VERIFICARE, VALIDARE ȘI TESTARE AUTOMATĂ

Prezentarea cursului
[05 Octombrie 2019]

Lector dr. Camelia Chisăliță-Crețu
Universitatea Babeș-Bolyai, NTTData
Programul Postuniversitar de Pregătire și Formare Profesională în Informatică

Conținut

- Activitățile asociate cursului
- Planificarea activităților
- Obiectivele cursului
- Laborator
- Evaluare
 - Examen scris. Tipuri de subiecte
 - Proiect Test Automation
- Resurse web
- Referințe bibliografice

Activități asociate

- **Curs:** 4 ore/săptămână;
- **Laborator:** 4 ore/săptămână;
- **Durata:** 7 săptămâni.

Planificarea activităților (1)

Săptămâna	Durata	Curs	Laborator
[01]	4 ore	Prezentare curs Curs 01. Verificare și Validare	Lab 01. Instalare Java, IntelliJ IDEA Setup proiect JUnit
		Curs 02. Inspectare	
[02]	4 ore	Curs 03. Testare	Lab 02. Configurare git Inspectare
		Curs 04. Testare Black-Box	
[03]	4 ore	Curs 05. Testare White-Box	Lab 03. Testare Black-Box
		Curs 06. Testing Management Tools. Demo TestLink	
[04]	4 ore	Curs 07. Niveluri de testare	Lab 04. Testare White-Box Coverage Tools
		Curs 08. Continuous Integration. Demo Jenkins + git + Maven	

Planificarea activităților (2)

Săptămâna	Durata	Curs	Laborator
[05]	4 ore	Curs 09. Testare automata	Lab 05. Test Management Continuous integration (TestLink, Jenkins)
		Curs 10. Automation Testing Tools. Demo Selenium WebDriver + Serenity BDD	
[06]	4 ore	Curs 11. Corectitudinea programelor. Floyd. Hoare	Lab 06. Testare automata [1]. Selenium WebDriver + Serenity BDD
		Curs 12. Corectitudinea programelor. Dijkstra	
[07]	4 ore	Curs 13. Raportarea bug-urilor (RIMGEN)	Lab 07. Testare automata [2]. Selenium WebDriver + Serenity BDD
		Curs 14. Recapitulare și pregătire pentru examen	
[08]	Sesiune: Examen scris/ Susținere proiect Test Automation		
[09]	Sesiune restanțe:Examen scris/ Susținere proiect Test Automation		

Obiectivele cursului

- Studiarea activităților legate de verificare și validare:
 - Studiarea metodelor de **verificare statică și dinamică** a sistemelor soft [[Dromey1989](#), [Frențiu2010](#)].
 - Inspectarea programelor [[Myers2004](#)];
 - Proiectarea cazurilor de testare folosind metodele de testare black-box și white-box [[Myers2004](#), [NT2005](#), [Crosby1980](#), [Juran1998](#), [Weinberg1992](#), [Pressman2000](#), [BBST2008](#), [Patton2005](#), [Frențiu2010](#)];
 - Utilizarea tool-urilor în testare;
 - Înțelegerea noțiunilor: **algoritm parțial și total corect, terminarea algoritmului** [[Frențiu2010](#)];
 - Proiectarea algoritmilor în paralel cu demonstrarea corectitudinii lor [[Hoare1969](#), [Dijkstra1975](#), [Bălănescu1995](#)];
 - Dezvoltarea corectă a programelor din specificații [[Dijkstra1968](#)].

Laborator

- Aplicarea conceptelor discutate la curs la nivelul unui proiect particular;
- Teme de laborator:
 - Lab 01. Setup proiect. JUnit (**5%**);
 - Lab 02. Inspectare. Git (**10%**);
 - Lab 03. Testare Black-box (**15%**);
 - Lab 04. Testare White-box (**15%**);
 - Lab 05. Test Management Tools. Continuous Integration (**25%**);
 - Lab 06. Testare automată [1];
 - Lab 07. Testare automată [2] (**30%**).
- Fiecare săptămână de întârziere se penalizează cu 2 puncte din nota acordată.
- **L = media ponderată a punctelor acordate temelor de laborator**

Evaluare

- Studenții pot opta pentru una dintre următoarele forme de evaluare în sesiunea de examene și/sau sesiunea de restanțe:
 - Examen scris (E);
 - Proiect Test Automation (PTA).
- **Studenții pot alege forma de evaluare până Sâmbătă, 16 Noiembrie 2019, orele 12:00 (ultima săptămână din modul).**
 - Studenții care nu au optat pentru nici una dintre formele de evaluare, vor susține examen scris.
- Nota finală se obține astfel:

Nota finală = 50% L + 50% E sau PTA

Examen scris. Tipuri de subiecte

1. Evaluare în sesiunea de examen prin **examen scris**, fără resurse (*books closed*);
2. Tipuri de subiecte:
 - a. **scurt (S)** – complexitate redusă, cu o rezolvare de max. o (1) pagină;
 - b. **lung (L)** – complexitate medie sau ridicată, cu o rezolvare de max. două (2) pagini;
3. Lista subiectelor posibile (*scurte si lungi*) este disponibilă la adresa [Pregătire examen](#).
4. Pe baza listei inițiale de subiecte *scurte* și *lungi*, fiecare student va elabora o listă proprie cu 7 subiecte scurte și 7 subiecte lungi ("7S+7L"). **Din această listă, în ziua examenului scris, studenții vor primi două (2) subiecte scurte și un (1) subiect lung;**
5. Lista proprie 7S+7L se completează de fiecare student online, la adresa [Pregătire examen](#).
6. **Termen de completare a listei proprii 7S+7L: până cel târziu cu 7 zile înainte de examenul scris.**
7. Lista subiectelor posibile pentru examenul scris este actualizată după fiecare curs.
8. Durata examenului scris este 1h30min.

$$E = 20\% \text{SubiectScurt} + 20\% \text{SubiectScurt} + 50\% \text{SubiectLung} + 1 \text{ punct Oficiu.}$$

Proiect Test Automation

1. Proiectul Test Automation (PTA) înlocuiește susținerea examenului scris.
2. PTA este un proiect individual, nu poate fi susținut în echipe formate din mai mulți studenți.
3. Pentru elaborarea și susținerea proiectului se poate înscrie individual, orice student.
4. Înscrierea constă în completarea datelor referitoare la cerințele de realizare (vezi [StudentiPTA](#)).
5. **Termenul limită pentru înscriere este 16 Noiembrie 2019, orele 12:00.**
6. **Un student care s-a înscris pentru susținerea PTA poate renunța și va susține examenul scris Sâmbătă, 23 Noiembrie 2019.**
7. Susținerea PTA va avea loc în aceeași zi în care are loc examenul scris. Planificarea susținerilor va fi disponibilă cu o zi înainte de examenul scris.
8. Susținerea constă în:
 - prezentarea particularităților tool-ului;
 - prezentarea arhitecturii aplicației de testare;
 - rularea propriu-zise a testelor;
 - efectuarea unor modificări minimale și apoi rularea testelor.
9. **Timpul de prezentare a PTA alocat unui student este de max. 15 minute.**

Resurse web

- Slack
 - link: slack.com
 - workspace: [VVTA2019](#)
- Link-uri către fișiere disponibile pe Google Drive
 - [Pregatire examen](#)
 - [StudentiPTA](#)
- Tool-uri/Platforme care folosesc contul SCS
 - **IntelliJ IDEA** (Ultimate, pentru obtinerea unei licente de folosire gratuita);
 - **BitBucket** (activare cont);
 - **TestLink** (activare cont);
 - **Jenkins** (conectare).

Referințe bibliografice

- **[Frențiu2010]** M. Frențiu, Verificarea și validarea sistemelor soft, Presa Universitară Clujeană, 2010.
- **[Myers2004]** G. Myers, The Art of Software Testing, 2nd Edition, John Wiley, 2004.
- **[NT2005]** K. Naik and P. Tripathy. *Software Testing and Quality Assurance*, Wiley Publishing, 2005.
- **[Crosby1980]** Philip B. Crosby, *Quality Is Free*, Signet Shakespeare, 1980.
- **[Juran1998]** A. Blanton Godfrey, Joseph Juran, *JURANS QUALITY HANDBOOK*, McGraw-Hill, 1998.
- **[Weinberg1992]** Gerald Weinberg, *Quality Software Management , Vol. 1: Systems Thinking*, Dorset House Publishing, 1992.
- **[Pressman2000]** Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, McGraw-Hill, Inc., 2000.
- **[BBST]** BBST – Bug Advocacy Course,
<http://testingeducation.org/BBST/>(<http://testingeducation.org/BBST/bugadvocacy/BugAdvocacy2008.pdf>).
- **[Patton2005]** R. Patton, *Software Testing*, Sams Publishing, 2005.
- **[Dromey1989]** G. Dromey, Program Derivation. The Development of Programs From Specifications, Addison Wesley Publishing Company, 1989.
- **[Bălănescu1995]** T. Bălănescu, Corectitudinea programelor, Editura tehnică, București, 1995.
- **[Dijkstra1068]** E. Dijkstra, A constructive approach to the problem of program correctness, BIT, 8:174–186, 1968.
- **[Hoare1969]** C.A.R. Hoare, An axiomatic basis for computer programming, CACM, 12(583):576–580, 1969.