

VERIFICAREA ȘI VALIDAREA SISTEMELOR SOFT

Prezentarea cursului
[26 Februarie 2024]

Lector dr. Camelia Chisăliță-Crețu
Universitatea Babeș-Bolyai Cluj-Napoca

Conținut

- Activitățile cursului
- Obiectivele cursului
- Planificarea activităților pe săptămâni
- Platforme online
- Seminar. Reguli. Evaluare. Tematică
- Laborator. Reguli. Evaluare. Tematică
- Examen scris
- Evaluare finală
- Referințe bibliografice

Activitățile cursului

- **disciplină:** obligatorie;
- **evaluare:** examen în sesiune;
- **desfășurare:** 12 săptămâni;
- **număr credite:** 7;
- **curs:** 2 ore/săptămână;
- **seminar:** 1 oră/săptămână;
- **laborator:** 1 oră/săptămână;
- **syllabus:** [link](#)

Obiectivele cursului

- Studierea activităților legate de verificare și validare:
 - **Metode de verificare statică și dinamică** a sistemelor soft [[Dromey1989](#), [Frențiu2010](#)].
 - **Inspectarea documentelor** [[Myers2004](#)];
 - **Testare.** Niveluri de testare. Tipuri de testare. Tehnici de testare. Abordări în testare (BBT, WBT) [[Myers2004](#), [NT2005](#), [Crosby1980](#), [Juran1998](#), [Weinberg1992](#), [Pressman2000](#), [BBST2008](#), [BBST2010](#), [BBSS2011](#), [Patton2005](#), [Frențiu2010](#), [TryQA2023](#), [Firesmith2015](#), [Easterbrook2010](#), [Altom2016](#), [Draghia2023](#)];
 - Utilizarea instrumentelor asociate diferitelor activități de testare (management, execuție, raportare, etc.);
 - Execuția simbolică [[PV2008](#)];
 - Verificarea modelelor asociate sistemelor soft [[KB2008](#)].
 - **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](#)].

Planificarea activităților pe săptămâni (1)

Săptămâna	Curs	Seminar	Laborator
[S01] 26 februarie – 01 martie	Curs 01. Introducere în verificare și validare. Inspectare	Seminar 01. Inspectare	Lab 01. Inspectare. Maven. Git. SonarLint
[S02] 04 martie – 08 martie	Curs 02. Testare Black-Box		
[S03] 11 martie – 15 martie	Curs 03. Testare White-Box	Seminar 02. Testare Black-Box	Lab 02. Testare Black-Box. TestLink
[S04] 18 martie – 22 martie	Curs 04. Niveluri de testare		

Planificarea activităților pe săptămâni (2)

Săptămâna	Curs	Seminar	Laborator
[S05] 25 martie – 29 martie	Curs 05. [Evozon Systems]: Test Automation. Performance Testing	Seminar 03. Testare White-Box	Lab 03. Testare White-Box. Test Coverage Tools
[S06] 01 aprilie – 05 aprilie	Curs 06. Instrumente utilizate in testare (Mockito. Testink. Jenkins)		
[S07] 08 aprilie – 12 aprilie	Curs 07. [Altom]: Mituri în testare	Seminar 04. Niveluri de testare	Lab 04. Niveluri de testare. Jenkins + TestLink
[S08] 15 aprilie – 19 aprilie	Curs 08. Corectitudine. Floyd. Hoare		

Planificarea activităților pe săptămâni (3)

Săptămâna	Curs	Seminar	Laborator
[S09] 22 aprilie – 26 aprilie	Curs 09. [FundMore.ai]: Quality Assurance. Quality Control	Seminar 05. Corectitudine (Floyd)	Lab 05. Testare automată. Selenium WebDriver. Serenity BDD
[S10] 29 aprilie – 03 mai	Curs 10. Execuție simbolică	Seminar 05. Corectitudine (Floyd)	
[S--] 06 mai – 12 mai	Vacanța de Paști		
[S11] 13 mai – 17 mai	Curs 11. Raportarea bugurilor	Seminar 06. Rafinarea algoritmilor	Lab 06. Finalizarea activității de laborator
[S12] 20 mai – 24 mai	Curs 12. Recapitulare + Simulare examen		

Planificarea activităților pe săptămâni (3)

Săptămâna	Curs	Seminar	Laborator
[S--] 27 mai – 09 iunie	Sesiune de examene (2 săptămâni)		
[S--] 10 iunie – 16 iunie	Sesiune de restanțe (1 săptămână)		
[S--] 17 iunie – 30 iunie	Sesiune de lichidate (1 săptămână) Pregătirea examenului de licență (2 săptămâni)		

Platforme online

- **MS Teams**
 - studenții sunt înscriși implicit în team-ul **VVS2024**;
- channels:
 - **General**
 - evenimente programate, anunțuri;
 - acces la **Class Materials** (materiale curs, fișiere de prezență, teme de lab, activități de seminar, tutoriale, etc.);
 - **Lectures**
 - prezentarea cursului, notițe de curs, bibliografie;
 - **Seminars**
 - activitatea de seminar pentru toate grupele;
 - **Labs**
 - discuții și clarificări legate de temele de laborator;
 - **alte channel-uri**
 - Dedicat activităților specifice de la seminar.

Evaluare finală

- **evaluare finală:**
 - **$NF = 50\%L + 10\%S + 40\%E$**
- **condiții de promovare** în sesiunea normală și sesiunea de restanțe, considerând că studentul îndeplinește **condițiile de participare** la examen (**5 prezențe lab + 4 prezențe seminar**):
 - examen scris (**E**): $E \geq 1.00$;
 - **$NF \geq 5.00$.**
- [Link fișier pentru calcularea NF](#)

Seminar. Reguli. Evaluare

- prezența:
 - **prezența la activitățile de seminar este obligatorie (min. 4 prezențe din 6 posibile);**
 - se notează conform temelor planificate. De exemplu, prezența pentru Seminar01 se va nota doar în S01 și S02. Participarea ulterioară, la mai multe activități de seminar similare, cu intenția de a recupera prezența pentru Seminar01 nu va fi luată în considerare.
- evaluare activitate de seminar:
 - **S** = I_{sem} , unde
 - implicarea în cadrul activității de seminar:
 - $I_{sem} = \frac{10}{6} \sum_{j=1}^6 i_j$, i_j - implicarea la seminarul j.
- **pondere (S): 10% din nota finală (NF).**

Seminar. Tematică

- **obiectiv:**
 - aplicarea conceptelor discutate la curs la nivelul unor probleme/aplicații concrete.
- **teme de seminar:**
 - **Seminar 01.** Inspectare;
 - **Seminar 02.** Testare black-box;
 - **Seminar 03.** Testare white-box;
 - **Seminar 04.** Niveluri de testare;
 - **Seminar 05.** Corectitudine. Metoda lui Floyd;
 - **Seminar 06.** Corectitudine. Dezvoltarea algoritmilor din specificații (Rafinarea algoritmilor).

Laborator. Reguli

1. **Prezența la activitatea de laborator este obligatorie (min. 5 prezențe din 6 posibile).**
2. Termenul de predare pentru fiecare temă de laborator este următorul laborator. Întârzierea față de termenul inițial se depunțează cu **2 puncte** față de primul termen de predare.
3. După ultimul termen de predare, fiecare temă de laborator nepredată se notează cu 0.
4. Temele de laborator se pot preda **doar** în cadrul laboratorului semigrupei în care studentul s-a înscris la începutul semestrului.
5. **Studentii restanțieri sau în prelungire de studii trebuie să participe la activitatea de laborator planificată în acest an universitar. Nu se vor lua în considerare activitățile de laborator desfășurate în anii universitari precedenți.**

Laborator. Tematică

- **obiectiv:**
 - aplicarea conceptelor discutate la curs, la nivelul unui proiect particular.
- **teme de laborator:**
 - **Lab 01.** Inspectare. Proiect Maven. Git. SonarLint (**10%**);
 - **Lab 02.** Testare black-box. Repository Git. TestLink (**15%**);
 - **Lab 03.** Testare white-box. Test Coverage Tools (**15%**);
 - **Lab 04.** Testare de integrare. Jenkins + TestLink (**30%**);
 - **Lab 05.** Testare automată. Selenium WebDriver + Serenity BDD (**30%**).

Laborator. Evaluare

- prezența:
 - **prezența la activitățile de laborator este obligatorie (min. 5 prezențe din 6 posibile);**
 - se notează conform temelor planificate. De exemplu, prezența pentru Lab01 se va nota doar în S01 și S02. Participarea ulterioară la mai multe activități de laborator similare cu intenția de a recupera prezența pentru Lab01 nu va fi luată în considerare.
- evaluare activitate de seminar:
 - $L = N_{Lab}$, unde
 - evaluarea temelor de laborator:
 - $N_{Lab} = \frac{1}{100} \sum_{i=1}^5 L_i * pondereL_i$, unde
 - L_i = nota acordată temei de laborator $i, i = \overline{1, 5}$;
 - $pondereL_i$ = ponderea temei de laborator $i, i = \overline{1, 5}$;
 - $pondereL[] = \{10\%, 15\%, 15\%, 30\%, 30\%\}$.
 - **pondere (L): 50% din nota finală (NF).**

Examen scris

- **condiții de participare** la examenul scris (**E**) din sesiunea normală și sesiunea de restanțe:
 - **numărul de prezențe la laborator ≥ 5** (90% din numărul de laboratoare);
 - **numărul de prezențe la seminar ≥ 4** (75% din numărul de seminarii);
 - $L \geq 0.00$, $S \geq 0.00$;
 - **în sesiunea de restanțe nu se predau teme de laborator.**
- **desfășurare:** fizic (scris).
- **durata:** ~45minute;
- **subiect (max. 10 puncte):**
 - întrebări grilă (9 puncte);
 - oficiu (1 punct);
- **tematică de examen:** se indică la finalul fiecărui curs.
- **pondere (E): 40% din nota finală (NF).**

Evaluare finală

- **evaluare finală:**
 - **$NF = 50\%L + 10\%S + 40\%E$**
- **condiții de promovare** în sesiunea normală și sesiunea de restanțe, considerând că studentul îndeplinește **condițiile de participare** la examen (**5 prezențe lab + 4 prezențe seminar**):
 - examen scris (**E**): $E \geq 1.00$;
 - **$NF \geq 5.00$.**
- [Link fișier pentru calcularea NF](#)

Referințe bibliografice (1)

- **[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.
- **[PV2008]** C. Pasareanu și W. Visser, Symbolic Execution and Model Checking for Testing, Springer, 2008.
- **[KB2008]** J.P. Katoen și C. Baier, Principle of Model Checking, MIT Press Cambridge, London, UK, 2008.
- **[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.

Referințe bibliografice (2)

- **[Hoare1969]** C.A.R. Hoare, An axiomatic basis for computer programming, CACM, 12(583):576–580, 1969.
- **[BBST2010]** Black-Box Software Testing (BBST), Foundations, <http://www.testingeducation.org/BBST/foundations/BBSTFoundationsNov2010.pdf>.
- **[BBST2011]** BBST – Test Design, Cem Kaner, <http://www.testingeducation.org/BBST/testdesign/BBSTTestDesign2011pfinal.pdf>.
- **[Firesmith2015]** Donald Firesmith, *Four Types of Shift Left Testing*, https://insights.sei.cmu.edu/sei_blog/2015/03/four-types-of-shift-left-testing.html
- **[Easterbrook2010]** S. Easterbrook, *Software Testing*, <http://www.easterbrook.ca/steve/2010/11/the-difference-between-verification-and-validation/>.
- **[TryQA2023]** ISTQB Exam Certification, <http://istqbexamcertification.com/what-is-a-defect-life-cycle/>.
- **[Altom2016]** Levente Balint, *BLOG : RIMGEN, How Well Do you Advocate For Your Bugs?*, <http://altom.training/blog/tag/rimgen/>.
- **[Draghia2023]** Claudiu Draghia, <http://testingchallenges.thetestingmap.org/>.
- **[PLGM2022]** Sheena Panthaplackel, Junyi Jessy Li, Milos Gligoric, Raymond J. Mooney, Learning to Describe Solutions for Bug Reports Based on Developer Discussions, ACL 2022, pp. 2935 – 2952.