

FIȘA DISCIPLINEI

Verificarea și validarea sistemelor soft

Anul universitar 2025-2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai
1.2. Facultatea	Facultatea de Matematică și Informatică
1.3. Departamentul	Departamentul de Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Informatică (limba Română)
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Verificarea și validarea sistemelor soft				Codul disciplinei	MLR5014
2.2. Titularul activităților de curs		Lector dr. Maria-Camelia Chisăliță-Crețu					
2.3. Titularul activităților de seminar		Lector dr. Maria-Camelia Chisăliță-Crețu					
2.4. Anul de studiu	3	2.5. Semestrul	6	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Obligativu

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Număr de ore pe săptămână	4	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	2
3.4. Total ore din planul de învățământ	48	din care: 3.5. curs	24	3.6 seminar/laborator/proiect	24
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					35
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					46
Tutoriat (consiliere profesională)					6
Examinări					6
Alte activități: comunicare bidirecțională cu titularul de disciplină					4
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				127	
3.8. Total ore pe semestru				175	
3.9. Numărul de credite				7	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	• Programare orientată obiect, Metode avansate de programare, Medii de proiectare și programare, Programare Web
4.2. de competențe	• Cunoașterea mediilor de programare pentru dezvoltarea aplicațiilor în limbaje orientate-obiect de nivel înalt

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	• Sală de curs cu videoproiector
5.2. de desfășurare a seminarului/ laboratorului	• Laborator cu calculatoare conectate la Internet, servere Web pentru găzduirea site-urilor Web și aplicațiilor Web dezvoltate pe baza tehnologiilor PHP, Java, .NET.

6.1. Competențele specifice acumulate¹

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

Competențe profesionale / esențiale	- dezvoltarea și întreținerea aplicațiilor informatice - utilizarea instrumentelor informatice în context interdisciplinar
Competențe transversale	- aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională - desfășurarea eficientă a activităților organizate într-un grup interdisciplinar și dezvoltarea capacităților empatică de comunicare interpersonală, de relaționare și colaborare cu grupuri diverse

6.2. Rezultatele învățării

Cunoștințe	Studentul cunoaște metodele de testare și verificare a sistemelor software. Studentul este familiarizat cu instrumente folosite pentru testarea, depanarea, validarea aplicații software. Studentul este familiarizat cu instrumentele de management de proiect, sistemele de control al versiunilor, precum și conceptele, metodele, instrumentele de continuous integration/continuous delivery (CI/CD).
Aptitudini	Studentul are abilitatea de a alege și folosi module și medii existente pentru dezvoltarea de aplicațiilor. Studentul este capabil să aplice șabloane arhitecturale, șabloane de proiectare și bunele practici în domeniu pentru a proiecta aplicații software de complexitate mare. Studentul are abilitatea de a crea teste automate de diferite nivele de granularitate pentru asigurarea calității sistemelor dezvoltate.
Responsabilități și autonomie	Studentul are capacitatea de a observa și obține informații din diverse surse. Studentul are capacitatea de a identifica probleme complexe și să examineze probleme conexe pentru a dezvolta opțiuni de rezolvare și implementa soluții. Studentul are capacitatea de a evalua diferite arhitecturi și soluții posibile pentru o problema și a alege pe cel potrivit pentru cerințele și constrângerile specifice aplicației de dezvoltat.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	- Înțelegerea noțiunilor de algoritm parțial și total corect. - Formarea deprinderilor de proiectare a algoritmilor în paralel cu demonstrarea corectitudinii lor. - Cunoașterea metodelor de testare și verificare a sistemelor soft. - Formarea deprinderilor de proiectare a programelor corecte din specificații. - Formarea unui stil modern de programare.
7.2 Obiectivele specifice	- Studentii vor ști cum se desfășoară și care sunt pașii unei inspectări din fiecare etapă de dezvoltare a sistemului soft. - Studentii vor ști să prevadă încă din faza de specificare și proiectare crearea unor cazuri de testare care să îi ajute la dezvoltarea unui sistem soft robust. - Studentii vor ști să utilizeze instrumentele pentru managementul procesului de testare. - Studentii vor ști să proiecteze cazurile de testare folosind diferite criterii (black-box, white-box).

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Verificarea și validarea sistemelor soft. Inspectarea programelor	Prezentare, Demonstrație, Problematizare	
2. Testarea programelor (1): Conceptul de testare a programelor. Criterii de testare. Testarea black-box.	Prezentare, Demonstrație, Problematizare	
3. Testarea programelor (2): Criterii de testare. Testarea white-box.	Prezentare, Demonstrație, Problematizare	
4. Niveluri de testare. Tipuri de testare.	Prezentare, Demonstrație, Problematizare	
5. Tehnici de testare. Instrumente de testare automată.	Prezentare, Demonstrație, Problematizare	
6. Testarea aplicațiilor Web. Selenium WebDriver și Serenity BDD	Prezentare, Demonstrație, Problematizare	
7. Teoria corectitudinii programelor. Contribuțiile lui Floyd, Hoare. Contribuțiile lui Dijkstra.	Prezentare, Demonstrație, Problematizare	
8. Instrumente folosite în testarea unitară și testarea de integrare	Prezentare, Demonstrație, Problematizare	
9. Raportarea bug-urilor	Prezentare, Demonstrație, Problematizare	
10. Asigurarea calitatii produselor soft. Controlul calitatii	Prezentare, Demonstrație, Problematizare	
11. Executie simbolică	Prezentare, Demonstrație, Problematizare	
12. Verificarea modelelor Pregătire pentru examen	Prezentare, Demonstrație, Problematizare	

Bibliografie

Cărți

1. Frentiu, M., Verificarea și validarea sistemelor soft, Presa Universitară Clujeană, 2010
2. R. S. Pressman, Software engineering: a practitioner's approach, seventh edition, Higher Education, 2010
3. L. Crispin, J. Gregory, Agile testing: a practical guide for testers and agile teams, Addison-Wesley, 2009
4. M. Pezzand, M. Young, Software Testing and Analysis: Process, Principles and Techniques, John Wiley & Sons, 2008
5. K. Naik, P. Tripathy, Software testing and quality assurance. Theory and Practice, A John Wiley & Sons, Inc., 2008
6. J. P. Katoen, Principles of Model Checking, MIT Press, May 2008
7. R. Patton, Software Testing, Sams Publishing, 2005
8. Glenford J. Myers, The Art of Software Testing, John Wiley & Sons, Inc., 2004
9. I. Bernstein, Practical software testing, Springer, 2002
10. Balanescu T., Corectitudinea programelor, Editura tehnica, București 1995.
11. Morgan, C., Programming from Specifications, Prentice Hall, New York, 1990.
12. Dromey G., Program Derivation. The Development of Programs From Specifications, Addison Wesley Publishing Company, 1989.
13. Heusser, M., Larsen, M. (2023), Software Testing Strategies: A Testing Guide for the 2020s, United Kingdom: Packt Publishing.

Articole

1. Hoare, C.A.R., An axiomatic basis for computer programming, CACM, 12(1969), pg.576-580, 583.
2. Dijkstra, E., A constructive approach to the problem of program correctness, BIT, 8(1968), pg.174-186.
3. Dijkstra, E., Guarded commands, nondeterminacy and formal derivation of programs, CACM, 18(1975), 8, pg.453-457.
4. 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.

Tutoriale

Pentru activitățile curs/seminar/laborator se vor furniza tutoriale adaptate cerințelor temelor asociate.

8.2 Seminar / laborator	Metode de predare	Observații
S1: Inspectare L1: Inspectare Inspection tools. Issue tracker tools. Test management tools	Prezentare, conversatie, Problematizare, Descoperire, Studiu individual, Exercitii	
S2: Specificarea cazurilor de testare pe baza specificatiilor (BBT) L2: Proiectarea cazurilor de testare pe baza specificatiilor (BBT) Test management tools. Continuous Integration tools	Prezentare, conversatie, Problematizare, Descoperire, Studiu individual, Exercitii	
S3: Specificarea cazurilor de testare pe baza codului sursa (WBT) L3: Proiectarea cazurilor de testare pe baza codului sursa (WBT) Test management tools. Continuous Integration tools	Prezentare, conversatie, Problematizare, Descoperire, Studiu individual, Exercitii	
S4: Niveluri de testare L4: Niveluri de testare Test management tools. Continuous Integration tools	Prezentare, conversatie, Problematizare, Descoperire, Studiu individual, Exercitii	
S5: Corectitudine. Floyd L5: Testare GUI/Web Web testing tools. Test management tools. Continuous Integration tools	Prezentare, conversatie, Problematizare, Descoperire, Studiu individual, Exercitii	
S6: Rafinare din specificatii L6: Analiza statica: JML, ESC2Java	Prezentare, conversatie, Problematizare, Descoperire, Studiu individual, Exercitii	
Bibliografie <i>similă cu bibliografia pentru curs</i>		

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

- Conținutul disciplinei este în concordanță cu recomandările ACM pentru domeniul informatică.
- Conținutul cursului este inclus în programele de studii ale universităților de prestigiu din străinătate.
- Conținutul cursului este considerat relevant de către companiile software care activează în domeniul dezvoltării software și utilizează testarea software.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Se va acorda nota E pentru examinarea scrisa.	Examen scris în sesiune	40%
10.5 Seminar/laborator	Activitarea de seminar, constând în participarea la discuții și la rezolvarea unor exerciții va fi notată cu o nota S .	Activitate de Seminar	10%
	Activitatea de laborator va fi notată cu nota L .	Teme de laborator	50%
Observații:			
• Activitatea de laborator nu se reevalua în timpul sesiunilor (de examinare sau reexaminare). • Activitatea de laborator pentru studenții restanțieri se reface conform cerințelor curente.			

- Prezența la activitatea de laborator se poate face doar cu grupa din studentul face parte.
- Predarea cu întârziere duce la diminuarea cu 2 puncte din nota acordată.
- După depășirea termenului de predare, tema de laborator va fi notată cu 0.
- Condiții de participare la examenul final: 75% la activitățile de seminar (minim 4 prezențe) și 90% la activitățile de laborator (minim 5 prezențe).

10.6 Standard minim de performanță

- Studenții vor ști să utilizeze instrumentele pentru managementul procesului de testare.
- Studentii vor ști să aplice diferite metode de testare asupra produselor soft.
- Studenții vor fi capabili să proiecteze cazurile de testare folosind diferite criterii (black-box, white-box).
- Studenții vor avea cunoștințe teoretice referitoare la diverse metode de verificare (inspectare, testare, demonstrarea corectitudinii).
- Media finală (M) se obține astfel: $M = 40\%E + 10\%S + 50\%L$.
- Promovarea disciplinei presupune obținerea $M \geq 5,00$, unde $S \geq 0,00$, $L \geq 0,00$, $E \geq 0,00$.

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

Nu se aplică.

Data completării:

Semnătura titularului de curs

Semnătura titularului de seminar

15 aprilie 2025

Lector dr. Maria-Camelia CHISĂLIȚĂ-CREȚU Lector dr. Maria-Camelia CHISĂLIȚĂ-CREȚU

Data avizării în departament:

Semnătura directorului de departament

...

Conf. dr. Adrian STERCA

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "*Nu se aplică*".