

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	UNIVERSITATEA DIN ORADEA
1.2 Facultatea	INGINERIE ELECTRICĂ ȘI TEHNOLOGIA INFORMAȚIEI
1.3 Departamentul	CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
1.4 Domeniul de studii	CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
1.5 Ciclu de studii	LICENȚĂ
1.6 Programul de studii/Calificarea	CALCULATOARE

2. Date despre disciplină

2.1 Denumirea disciplinei	CALITATEA ȘI EVOLUȚIA SOFTWARE						
2.2 Titularul activităților de curs	Conf.univ.dr.inf. Moisi Elisa						
2.3 Titularul activităților de laborator	Asist.univ.drd.ing. Hannelore Sebestyen						
2.4 Anul de studiu	IV	2.5 Semestrul	VII	2.6 Tipul de evaluare	VP	2.7 Regimul disciplinei	O

(I) Impusă; (O) Opțională; (F) Facultativă

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

3.1 Număr de ore pe săptămână	4	din care: curs	2	laborator	2
3.4 Total ore din planul de învățământ	56	din care:curs	28	seminar/laborator	28
Distribuția fondului de timp ore					ore
Studiul după manual, suport de curs, bibliografie și notițe					10
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					8
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					14
Tutoriat					2
Examinări					10
Alte activități.....					
3.7 Total ore studiu individual	44				
3.9 Total ore pe semestru	100				
3.10 Numărul de credite	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Inginerie software
4.2 de competențe	

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Cursul se desfășoară față în față Sală de curs
5.2. de desfășurare a laboratorului	Prezența obligatorie la laborator 100% Laboratorul se desfășoară față în față Recuperarea laboratoarelor se face conform Regulament privind activitatea profesională a studenților în baza Sistemului European de Credite Transferabile (ECTS) Nivelul maxim admis al absențelor motivate sau nemotivate la activitățile practice (laboratoare, stagii, proiecte etc.) care pot fi recuperate este de până la 30% din totalul activităților. Sala de laborator dotată cu calculatoare în rețea, conexiune la internet și software adecvat

6.a. Competențele specifice acumulate

Competențe profesionale	C3. Definește cerințe tehnice C5. Remediază erori din software C6. Utilizează instrumente de analiză software asistată de calculator
-------------------------	--

Competențe transversale	
-------------------------	--

6.b. Rezultatele așteptate ale învățării

Cunoștințe	Studentul/absolventul descrie, identifică, sumarizează concepte și metode elementare privitoare la limbaje de programare, medii de programare, tehnici de programare și modul lor de aplicare în probleme concrete.
Aptitudini	Studentul/absolventul specifică cerințe, analizează, elaborează, dezvoltă și testează programe în limbaje de programare de uz general și /sau obiect-orientate, aplicând elementele specifice ingineriei software.
Responsabilitate și autonomie	Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei.

7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1 Obiectivul general al disciplinei	Își propune să furnizeze studenților instrumente și metode pentru evaluarea performanței sistemelor de calcul și software
7.2 Obiectivele specifice	Cursul vizează însușirea de către studenți a unor cunoștințe specifice performanțelor sistemelor de calcul, atât din punct de vedere software cât și hardware, a tehnicilor de evaluare prin măsurare, modelare analitică și simulare. Analiza și prezentarea datelor prin tehnici statistice. De asemenea se vizează prezentarea unor aspecte legate de performanța software.

8. Conținuturi

8.1 Curs	Metode de predare	Nr. Ore/ Observații
1. Introducere în validarea, verificarea și evoluția software-ului 2. Modele și standarde de calitate software 3. Fundamentele testării software 4. Framework-uri pentru testarea unităților și acoperirea testelor 5. Testarea de integrare și testarea de sistem 6. Testarea de regresie și automatizarea testelor 7. Analiza statică și dinamică, metrice și calitatea codului 8. Verificare intermediară și studii de caz privind eșecuri de validare 9. Modele de mentenanță software și legile lui Lehman 10. Refactorizare, reinginerie și datoria tehnică 11. Controlul versiunilor, strategii de ramificare și managementul lansărilor 12. Integrare continuă, livrare continuă și evoluția DevOps 13. Studii de caz în evoluția software-ului: open-source și industrie 14. Prezentările proiectelor finale și viitorul evoluției software-ului	expunerea orală și cu ajutorul videoproiectorului, dezbateri, răspunsuri la întrebări	28

Bibliografie

1. Ian Sommerville – Software Engineering, 10th Edition, Pearson, 2016.
2. Mauro Pezzè, Michal Young – Software Testing and Analysis: Process, Principles, and Techniques, Wiley, 2008.
3. Vaclav Rajlich – Software Engineering: The Evolution and Future of Software Development, CRC Press, 2016.

4. Martin Fowler – Refactoring: Improving the Design of Existing Code, 2nd Edition, Addison-Wesley, 2019.
5. Jez Humble, David Farley – Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Addison-Wesley, 2010.
6. Paul Ammann, Jeff Offutt – Introduction to Software Testing, 2nd Edition, Cambridge University Press, 2017.
7. Len Bass, Ingo Weber, Liming Zhu – DevOps: A Software Architect's Perspective, Addison-Wesley, 2015.
8. IEEE Std 829-2008 – Standard for Software Test Documentation.
9. ISO/IEC 25010:2011 – Systems and software engineering — Software product quality models.
10. Lehman, M. M. – Programs, Life Cycles, and Laws of Software Evolution, Proceedings of the IEEE, 1980.
11. Online Resources & Tools:
 - JUnit (Java): <https://junit.org>
 - PyTest (Python): <https://docs.pytest.org>
 - NUnit (C#): <https://nunit.org>
 - Selenium WebDriver: <https://www.selenium.dev>
 - Playwright Testing: <https://playwright.dev>
 - SonarQube: <https://www.sonarqube.org>
 - JaCoCo (Java coverage): <https://www.jacoco.org>
 - GitHub Actions CI/CD: <https://docs.github.com/en/actions>
 - GitLab CI/CD: <https://docs.gitlab.com/ee/ci/>
 - Jenkins: <https://www.jenkins.io>

8.2 Seminar		
8.3 Laborator	Metode de predare	Nr. Ore/Observații
1. Configurarea proiectului și familiarizarea cu baza de cod 2. Evaluarea atributelor de calitate software 3. Scrierea testelor unitare cu framework-uri 4. Analiza acoperirii și generarea rapoartelor 5. Testarea de integrare a componentelor sistemului 6. Testarea de regresie automatizată 7. Analiza statică și dinamică a calității codului 8. Prezentarea intermediară a validării proiectului 9. Cereri de modificare și corectarea erorilor 10. Refactorizare pentru îmbunătățirea mentenabilității 11. Ramificare în Git și simularea lansărilor 12. Integrare și livrare continuă 13. Minarea și analizarea depozitelor software 14. Prezentarea proiectelor finale	explicații, exemple, dialog	28

Bibliografie

1. Ian Sommerville – Software Engineering, 10th Edition, Pearson, 2016.
2. Mauro Pezzè, Michal Young – Software Testing and Analysis: Process, Principles, and Techniques, Wiley, 2008.
3. Vaclav Rajlich – Software Engineering: The Evolution and Future of Software Development, CRC Press, 2016.
4. Martin Fowler – Refactoring: Improving the Design of Existing Code, 2nd Edition, Addison-Wesley, 2019.
5. Jez Humble, David Farley – Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Addison-Wesley, 2010.
6. Paul Ammann, Jeff Offutt – Introduction to Software Testing, 2nd Edition, Cambridge University Press, 2017.
7. Len Bass, Ingo Weber, Liming Zhu – DevOps: A Software Architect's Perspective, Addison-Wesley, 2015.
8. IEEE Std 829-2008 – Standard for Software Test Documentation.
9. ISO/IEC 25010:2011 – Systems and software engineering — Software product quality models.
10. Lehman, M. M. – Programs, Life Cycles, and Laws of Software Evolution, Proceedings of the IEEE, 1980.
11. Online Resources & Tools:
 - JUnit (Java): <https://junit.org>
 - PyTest (Python): <https://docs.pytest.org>
 - NUnit (C#): <https://nunit.org>
 - Selenium WebDriver: <https://www.selenium.dev>
 - Playwright Testing: <https://playwright.dev>
 - SonarQube: <https://www.sonarqube.org>
 - JaCoCo (Java coverage): <https://www.jacoco.org>
 - GitHub Actions CI/CD: <https://docs.github.com/en/actions>
 - GitLab CI/CD: <https://docs.gitlab.com/ee/ci/>
 - Jenkins: <https://www.jenkins.io>

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

Conținutul disciplinei este în concordanță cu ceea ce se predă în alte centre universitare din străinătate

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- abilitatea de folosi bibliografia și de a urmări cursul - capacitatea de a aplica elementele teoretice la rezolvarea de probleme de natură practică	scris	50%
10.5 Seminar			
10.6 Laborator	- capacitatea de aplicare a cunoștințelor teoretice în rezolvarea problemelor practice - proiectarea și execuția testelor software, precum și utilizarea instrumentelor dedicate pentru evaluarea calității software	Laborator / lucrări practice – test practic	50%
10.7 Proiect			
Se va calcula un punctaj total, prin însumarea următoarelor punctaje întregi: • Punctaj 1: maxim 50 puncte: test grilă • Punctaj 2: maxim 50 puncte: test practic Media finală se va acorda pe baza curbei lui Gauss, conform regulilor/recomandărilor Universității			
10.8 Standard minim de performanță			
Standard minim de performanță: • explice conceptele de bază, • aplice cel puțin un framework și un instrument de testare, • utilizeze Git pentru operații simple, • recunoască rolul DevOps în evoluția software, • realizeze o lucrare practică ce demonstrează înțelegerea noțiunilor de testare și mentenanță. Aceste standarde pot fi atinse prin obținerea cumulativă a următoarelor punctaje: • cel puțin 50 puncte din totalul de 100 puncte din Punctaj 1-Punctaj 2; ○ cel puțin 25 puncte la Punctaj 1 ○ cel puțin 25 puncte la Punctaj 2			

Data completării
03.09.2025

Titular curs
Conf.univ.dr.inf. Elisa MOISI
emoisi@uoradea.ro

Titular laborator
Asist.univ.dr.drd.ing. Hannelore Sebestyen
hsebestyen@uoradea.ro

Data avizării în departament
24.09.2025

Semnătura directorului de departament:
Conf.univ.dr.inf. Elisa MOISI
emoisi@uoradea.ro

Data avizării în Consiliul Facultății IETI
25.09.2025

Semnătura Decan:
conf.univ.dr.ing.Eugen GERGELY

egergely@uoradea.ro