

FIȘA DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Petrol-Gaze din Ploiești
1.2. Facultatea	Inginerie Mecanică și Electrică
1.3. Departamentul	Automatică, Calculatoare și Electronică
1.4. Domeniul de studii universitare	Calculatoare și Tehnologia Informației
1.5. Ciclul de studii universitare	Licență
1.6. Programul de studii universitare	Calculatoare

2. Date despre disciplină

2.1. Denumirea disciplinei	Fiabilitate software
2.2. Titularul activităților de curs	Conf. univ. dr. ing. Pricop Emil
2.3. Titularul activităților seminar/laborator	Conf. univ. dr. ing. Pricop Emil
2.4. Titularul activității proiect	-
2.5. Anul de studiu	III
2.6. Semestrul *	2
2.7. Tipul de evaluare	V
2.8. Categoria formativă** / regimul*** disciplinei	DL/DFA

* numărul semestrului este conform planului de învățământ;

** DF - Discipline fundamentale; DS - discipline de specializare; DC - discipline complementare

*** obligatorie/impusă = DOB; opțională = DOP; facultativă = DFA

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

3.1. Număr de ore pe săptămână	3	din care: 3.2. curs	2	3.3. Seminar/laborator	1	3.4. Proiect	-
3.5. Total ore din planul de învățământ	42	din care: 3.6. curs	28	3.7. Seminar/laborator	14	3.8. Proiect	-
3.9. Total ore studiu individual (studiu după suport de curs, bibliografie și notițe, documentare suplimentară în bibliotecă, pe platformele electronice de specialitate, pregătire seminar/laboratoare, teme, referate, portofolii și eseuri)							8
3.10. Total ore pe semestru							50
3.11. Numărul de credite							2

4. Condiții (acolo unde este cazul)

4.1. de curriculum	<ul style="list-style-type: none"> ➤ Programarea calculatoarelor și limbaje de programare ➤ Programare orientată pe obiecte ➤ Instrumente pentru dezvoltarea programelor
4.2. de desfășurare a cursului	➤ Sală cu dotări multimedia (proiector / display smart)
4.3. de desfășurare a seminarului/laboratorului	➤ Laborator dotat cu sisteme PC în configurație standard pentru Windows, cu mediul de dezvoltare Visual Studio Community Edition, acces la internet și instalarea instrumentelor pentru testare și urmărirea defectelor (ex. NUnit, sistem de bug tracking)

5. Competențe specifice acumulate și rezultatele învățării* care stau la baza acestora

Competențe profesionale	Rezultatele învățării*
1. Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor	C1 – Studentul/absolventul descrie, identifică și sintetizează concepte și metode fundamentale referitoare la fiabilitatea software, la indicatorii de calitate ai produselor software, la tipurile de defecte și erori, la modelele de apariție a defectărilor și la metodele de analiză, testare și diagnoză utilizate pentru rezolvarea problemelor concrete din dezvoltarea și exploatarea sistemelor software.

	<p>A1 – Studentul/absolventul utilizează concepte, metode și instrumente specifice pentru analiza defectelor software, evaluarea indicatorilor de fiabilitate, proiectarea și aplicarea unor activități de testare și diagnoză</p> <p>RA1 – Studentul/absolventul manifestă spirit de inițiativă și responsabilitate în actualizarea continuă a cunoștințelor privind fiabilitatea, testarea și mentenanța software, adoptă o atitudine critică față de cauzele și efectele defectelor și utilizează în mod adecvat surse bibliografice și resurse profesionale relevante pentru creșterea calității produselor software.</p>
2. Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații	<p>C1 – Studentul/absolventul identifică, descrie și sintetizează concepte fundamentale privind performanța și fiabilitatea sistemelor software, explică relația dintre calitate, mentenabilitate și comportarea în exploatare și evidențiază factorii care influențează apariția defectelor, degradarea funcționării și necesitatea aplicării unor metode de creștere a fiabilității.</p> <p>A1 – Studentul/absolventul aplică metode și principii de bază pentru creșterea fiabilității software, prin utilizarea tehnicilor de testare, diagnoză, urmărire a defectelor, analiză a cauzelor și evaluare a mentenabilității, în scopul formulării unor soluții de îmbunătățire a calității și robusteții produselor software.</p> <p>RA1 – Studentul/absolventul arată spirit de inițiativă și acțiune în analiza critică a soluțiilor de creștere a fiabilității software, își asumă responsabilitatea pentru evaluarea și interpretarea rezultatelor obținute și promovează utilizarea unor practici de dezvoltare și mentenanță orientate spre calitate și reducerea defectelor.</p>
Competențe transversale	Rezultatele învățării*
1. Aplicarea, în mod responsabil, a principiilor, normelor și valorilor eticii profesionale în realizarea sarcinilor profesionale și identificarea obiectivelor de realizat, a resurselor disponibile, a etapelor de lucru, a duratelor de execuție, a termenelor de realizare aferente și a riscurilor aferente	<p>C1 – Studentul/absolventul identifică și descrie conceptele fundamentale din domeniul fiabilității software și al eticii profesionale, inclusiv aspecte privind responsabilitatea în dezvoltarea și testarea software, protecția proprietății intelectuale, utilizarea corectă a instrumentelor și datelor de test și raportarea onestă a defectelor și rezultatelor obținute.</p> <p>A1 – Studentul/absolventul identifică obiective, resurse și riscuri asociate activităților de analiză, testare și creștere a fiabilității software, elaborează scenarii de evaluare și argumentează alegerea metodelor utilizate prin raportare la cerințele de calitate, la constrângerile proiectului și la principiile eticii profesionale.</p> <p>RA1 – Studentul/absolventul are o comportare onorabilă, responsabilă și în spiritul legii în desfășurarea activităților specifice fiabilității software, respectă normele de etică academică și profesională și manifestă o atitudine responsabilă față de impactul defectelor software asupra utilizatorilor, organizațiilor și societății.</p>

* C – cunoștințe; A – aptitudini; RA – responsabilitate și autonomie.

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

6.1. Obiectivul general al disciplinei	<ul style="list-style-type: none"> ➤ La sfârșitul cursului studentul va fi familiarizat, prin cunoștințele dobândite, cu problematica dezvoltării de software cu grad înalt de fiabilitate și cu folosirea metodelor de analiză și remediere a defectelor software
6.2. Obiectivele specifice	<p>La sfârșitul cursului, studentul va fi capabil:</p> <ul style="list-style-type: none"> ➤ să înțeleagă conceptele fundamentale legate de fiabilitatea software; ➤ să caracterizeze principalele etape din ciclul de viață al unui produs software din perspectiva creșterii fiabilității; ➤ să cunoască și să utilizeze metodele și tehnicile de analiză și remediere a defectelor software.

7. Conținuturi

7.1. Curs	Nr. ore	Metode de predare	Observații
Calitatea sistemelor de programe. Indicatori de calitate. Relația între calitate, fiabilitate și mentenabilitate.	3	Interactivă și convențională, centrată pe student, folosind	

Concepte fundamentale în domeniul fiabilității software.	2	prezentări PowerPoint, demonstrații și exemple on-line	
Concepte de bază ale teoriei fiabilității (defecțiuni, redundanță, rata și intensificare defectărilor)	6		
Analiza erorilor în sistemele software	4		
Metode de testare și diagnoză a sistemelor software	4		
Mentenabilitatea. Factori și indicatori de mentenabilitate	3		
Metode de creștere a fiabilității software	4		
Sisteme software tolerante la defecte	2		
Bibliografie			
1. Roger K. Youree, Software Reliability Techniques for Real-World Applications, Ed. Wiley, 2023;			
2. Hoang Pham, System Software Reliability, Springer-Verlag, London, 2006.			
3. Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2015;			
4. Titu Băjenescu, Fiabilitatea sistemelor tehnice, MatrixRom, București, 2003;			
7.2. Seminar / laborator	Nr. ore	Metode de predare	Observații
Recapitularea cunoștințelor specifice pentru limbajele C++/C# și .NET	4	Interactivă și convențională, centrată pe student. Rezolvarea de exerciții.	
Utilizarea NUnit 3 pentru testarea software. Scrierea și rularea testelor unitare. Interpretarea rapoartelor.	4		
Sistem pentru urmărirea defectelor (ex. Bugzilla, Redmine, Jira) – raportarea, clasificarea și urmărirea bug-urilor	4		
Mentenabilitatea sistemelor software	2		
Bibliografie			
1. Roger K. Youree, Software Reliability Techniques for Real-World Applications, Ed. Wiley, 2023;			
2. Glenford Myers, Tom Badgett, Corey Sandler, The Art of Software Testing, Wiley, 2012			
3. Hoang Pham, System Software Reliability, Springer-Verlag, London, 2006.			
4. Documentația oficială NUnit – https://nunit.org			
5. Documentația oficială Bugzilla - https://www.bugzilla.org			
7.3. Proiect	Nr. ore	Metode de predare	Observații
<i>Nu este cazul</i>			
Bibliografie			
<i>Nu este cazul</i>			

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

- Conținuturile disciplinei sunt corelate cu recomandările și așteptările angajatorilor privind formarea competențelor de analiză a defectelor, testare, mentenanță și creștere a fiabilității produselor software, asigurând o abordare inginerescă, sistematică și orientată spre calitate a dezvoltării și întreținerii sistemelor informatice.

9. Evaluare

Tip activitate	9.1. Criterii de evaluare	9.2. Metode de evaluare	9.3. Pondere din nota finală
9.4. Curs	Cunoașterea noțiunilor fundamentale referitoare la fiabilitatea software, defecte comune (buguri), modalități de testare și remediere și indicatori de calitate software	Lucrare scrisă cu 30 de întrebări grilă sau cu răspuns scurt (30 întrebări x 0,3 pct. + 1 pct. oficiu)	60%
	Evaluarea implicării active la activitatea de curs	Cuantificarea numărului de prezențe la curs, participarea activă – răspunsuri la întrebări / rezolvarea de probleme la curs, etc.	10%
9.5. Seminar/laborator	Activitate laborator și verificări periodice	Evaluarea, atât pe parcursul semestrului a modului de realizare a lucrărilor de laborator, cât și la încheierea activității de laborator printr-o sesiune orală de întrebări și răspunsuri	30%
9.7. Standard minim de performanță			
<ul style="list-style-type: none"> ➤ Studentul trebuie să demonstreze cunoașterea noțiunilor de bază privind fiabilitatea software (cunoașterea indicatorilor de calitate, descrierea metodelor de testare a software-ului, cunoașterea metodelor de întreținere software). ➤ Obținerea a cel puțin 5 puncte la testul final cu 30 de întrebări. 			

Data
completării

19.09.2025

Semnătura titularului de curs
Conf. dr. ing. Pricop Emil

Semnătura titularului de
seminar/laborator
Conf. dr. ing. Pricop Emil

Semnătura titularului de
proiect

Data avizării în
departament

26.09.2025

Director de departament
Conf. dr. ing. Pricop Emil

Decan
Conf. dr. ing. Bădicioiu Marius