

FIȘA DISCIPLINEI

1. Date despre program

Facultatea	Facultatea de Inginerie Electrică și Știința Calculatoarelor
Departamentul	Departamentul de Calculatoare
Domeniul de studii	Inginerie electronică, telecomunicații și tehnologii informaționale
Ciclul de studii	Licență
Programul de studii	Rețele și software de telecomunicații

2. Date despre disciplină

Denumirea disciplinei	INGINERIE SOFTWARE PENTRU COMUNICAȚII				
Anul de studiu	IV	Semestrul	7	Tipul de evaluare	E
Regimul disciplinei	Categorია formativă a disciplinei DF - fundamentală, DD - în domeniu, DS - de specialitate, DC - complementară				DS
	Categorია de opționalitate a disciplinei: DO - obligatorie (impusă), DA - opțională (la alegere), DL - facultativă (liber aleasă)				DO

3. Timpul total estimat (ore alocate activităților didactice)

I a) Număr de ore pe săptămână	4	Curs	2	Seminar		Laborator	2	Proiect	
I b) Totalul de ore pe semestru din planul de învățământ	56	Curs	28	Seminar		Laborator	28	Proiect	

II Distribuția fondului de timp pe semestru:	ore
II.a) Studiu individual	66
II.b) Tutoriat (pentru ID)	
III. Examinări	3
IV. Alte activități (precizați):	

Total ore studiu individual II (a+b+c+d)	69
Total ore pe semestru (Ib+II+III+IV)	125
Numărul de credite	5

4. Competențe specifice acumulate

Competențe profesionale	CP2. Ajustează proiectele produselor CP4. Redactează rapoarte tehnice CP7. Identifică nevoile clientilor CP9. Stabilește procese de date CP14. Concepe designul produsului CP15. Sintetizează informații CP19. Elaborează proiecte de specificații pentru proiectare CP22. Interpretează cerințe tehnice
Competențe transversale	CT5. Promovează idei, produse sau servicii

5. Rezultatele învățării

Cunoștințe	Aptitudini	Responsabilitate și autonomie
Studentul/ absolventul trebuie:	Studentul/ absolventul trebuie:	La finalul disciplinei, studenții vor demonstra:
1. Să analizeze critic principiile, etapele și terminologia utilizate în ingineria programelor.	1. Să elaboreze specificații de cerințe software corecte și complete, utilizând tehnici și instrumente adecvate.	• Atitudine critică și responsabilă față de calitatea codului și a procesului de dezvoltare.
2. Să compare principalele metodologii de dezvoltare software și să	2. Să construiască modele software	• Capacitatea de a lucra în echipă

examineze modul de aplicare a conceptelor de management al proiectelor software.	prin diagrame UML care să descrie structura și comportamentul sistemului.	interdisciplinară, de a comunica eficient și de a documenta activitățile desfășurate.
3. Să explice procesul de inginerie a cerințelor și să adapteze conceptele fundamentale ale modelării software prin limbajul UML.	3. Să proiecteze arhitectura unui sistem software aplicând principii de modularitate și reutilizare.	• Deschidere către învățare continuă și adaptare la noi tehnologii, metodologii și instrumente software.
4. Să identifice și să aplice principii de proiectare, reutilizare și șabloane de proiectare software adecvate contextului unui proiect.	4. Să implementeze componente software conforme cerințelor definite, folosind instrumente moderne de dezvoltare.	• Conștientizarea importanței eticii și securității în dezvoltarea de produse software destinate utilizatorilor reali.
5. Să analizeze critic aspectele legate de testare, calitate și securitate în ciclul de viață al produselor software.	5. Să aplice tehnici și instrumente de testare pentru verificarea și validarea produselor software.	

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

Obiectivul general al disciplinei	Studentul/Absolventul trebuie să fie capabil să analizeze, să proiecteze și să implementeze soluții software complexe, de calitate, în conformitate cu cerințele considerate, utilizând metodologii adecvate și instrumente moderne, și să își asume responsabilitatea pentru calitatea, testarea și securitatea produselor program dezvoltate în echipă.
-----------------------------------	---

7. Conținuturi

Curs	Nr. ore	Metode de predare	Observații
1. Introducere, definiție, istoric, principii de bază, fazele ingineriei programării. Dicționar termeni	3	expunerea, prelegerea, conversație, studiu de caz, demonstrația	
2. Metodologii de dezvoltare software	3		
3. Managementul proiectelor software	2		
4. Ingineria cerințelor	3		
5. Modelarea sistemelor software			
5.1. Introducere, limbaje de modelare	2		
5.2. UML. Diagrame structurale. Exemple	2		
5.3. UML. Diagrame comportamentale. Exemple	2		
6. Proiectarea sistemelor software	2		
7. Implementarea sistemelor software	3		
8. Verificare și validare. Testare software. Tipuri de testare. Instrumente pentru managementul testării	3		
9. Inteligența artificială generativă și ingineria software	3		
Bibliografie minimală recomandată			
1. I. Sommerville, <i>Software Engineering</i> , 10th Edition, Pearson, India, 2017			
2. M. Richards and N. Ford, <i>Fundamentals of software architecture: An engineering approach</i> . Sebastopol, CA: O'Reilly Media, 2020.			
3. D. Farley, <i>Modern software engineering: Doing what works to build better software faster</i> . Boston, MA: Addison Wesley, 2022.			
4. Lara, Letaw. <i>Handbook of Software Engineering Methods</i> . 2024.			
5. *, <i>Software Engineering Body of Knowledge (SWEBOK)</i> , IEEE Computer Society, 2024, https://www.computer.org/education/bodies-of-knowledge/software-engineering			
6. Suportul electronic de curs			
7. Resurse online: tutoriale, articole de specialitate			

Aplicații (laborator)	Nr. ore	Metode de predare	Observații
1. Analiza stadiului actual în domeniul proiectului de diplomă/ Identificare și documentare tematică pentru a obține minim 2 direcții relevante pentru tema propusă. Familiarizarea cu un instrument specific de management al proiectelor	2	lucrări practice, exerciții, studii de caz, evaluare	
2. Elaborarea specificațiilor cerințelor software pentru proiectul de diplomă	2		
3. Proiectarea interfeței utilizator	2		
4. Proiectarea aplicației. Realizarea diagramelor UML	2		
5. Proiectare arhitectură II (diagrame UML: secvențe, stări, clase).	2		
6. Stabilire temă comună și echipe de lucru. Elaborarea specificațiilor cerințelor software pentru aplicația de realizat în echipă	2		
7. Proiectare wireframe UI. Proiectare Prototip UI.	2		
8. Proiectarea arhitectura. Realizarea diagramelor UML	2		
9. Planificare implementare versiunea MVP	2		
10. Implementare module aplicație utilizator final utilizand un CMS	4		
11. Elaborare plan de testare. Testare aplicație. Rafinare implementare. Pregătire aplicație pentru beneficiar	2		
12. Implementarea și execuția unui test automat pentru UI, pentru o aplicație oarecare. Testarea aplicației de către altă echipă	2		
13. Prezentare și evaluare aplicație și documentație elaborată	2		
Prezența la activitățile practice de laborator este obligatorie, conform regulamentelor USV în vigoare. Conform aceluiași regulamente, activitatea pe parcurs poate fi echivalată, la cerere, prin proiecte, pregătirea și participarea la concursuri profesionale, cu aprobarea cadrului didactic care conduce lucrările practice și cu condiția prezentării unui referat/ proiect, în care sunt descrise activitățile desfășurate și rezultatele obținute, cu evidențierea elementelor specifice tematicii disciplinei.			
Bibliografie minimală recomandată			
1. Resurse online: tutoriale, materiale bibliografice specifice limbajelor de programare utilizate, articole de specialitate			
2. I.Sommerville, Software Engineering, 10th Edition, Pearson India, 2017			
3. A. Cernian, A.D. Ioniță, Noțiuni aplicate de inginerie a sistemelor de programe, Ed. Matrixrom, București, 2009			
4. O. Gherghieș, A. Apetrei, Ingineria programării, Curs, Iași, 2002			
5. I. Odăgescu, F. Furtuna, Ingineria programării, Ed. ASE, București, 2002			
6. Fabian C., Mihalca R., Chichernea V., Goron S., Botezatu C., Iacob I., Proiectarea sistemelor informatice. Metode de realizare, Sylvi, Bucuresti 2001			
7. D. Bocu –Inițiere în ingineria sistemelor soft, Ed. Albastră, Cluj-Napoca, 2001			
8. Suportul electronic de laborator			

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Înțelegerea și cunoașterea noțiunilor fundamentale	Evaluarea cunoștințelor teoretice și practice din tematica studiată în timpul semestrului prin test grilă Evaluarea finală a aplicației practice elaborate.	30%
	Capacitatea de utilizare adecvată a noțiunilor din domeniul ingineriei software		20%
Laborator/ lucrări practice	Evaluarea capacității de analiză, sinteză și concretizare a cunoștințelor teoretice în identificarea soluțiilor problemei propuse la lucrările de laborator Corectitudinea și calitatea dezvoltării aplicației software și documentației aferente	Evaluare continuă a activităților desfășurate	50%

Fișa disciplinei include, dacă este cazul, elemente adaptate persoanelor cu dizabilități, în funcție de tipul și gradul acestora.

Data completării	Grad didactic, nume, prenume, semnătura titularului de curs	Grad didactic, nume, prenume, semnătura titularului de aplicație
22.09.2025	Prof. univ. dr.ing. Cristina TURCU	Ș.l.dr.ing. Felicia GÎZĂ

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
23.09.2025	Conf. dr. ing. Eugen COCA

Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
25.09.2025	Conf. dr. ing. Eugen COCA

Data aprobării în consiliul facultății	Grad didactic, nume, prenume, semnătura decanului
26.09.2025	Prof. univ. dr. ing. Laurențiu-Dan MILICI