

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Automatică și Calculatoare
1.3 Departamentul	Calculatoare
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Master
1.6 Programul de studii / Calificarea	Inginerie Software / Master
1.7 Forma de învățământ	IF – învățământ cu frecvență

2. Date despre disciplină

2.1 Denumirea disciplinei	Fiabilitatea programelor			Codul disciplinei	14
2.2 Titularul de curs	Prof. dr. ing. Suci Alin-Dumitru - alin.suciu@cs.utcluj.ro				
2.3 Titularul activităților de seminar / laborator / proiect	Prof. dr. ing. Suci Alin-Dumitru - alin.suciu@cs.utcluj.ro				
2.4 Anul de studiu	II	2.5 Semestrul	1	2.6 Tipul de evaluare	E
2.7 Regimul disciplinei	Categorii formative: <i>DF – fundamentală, DS - de specialitate, DC - complementară</i>				DS
	Opționalitate: <i>DOB - obligatorie, DOP - opțională, DFA - facultativă</i>				DOB

3. Timpul total estimat

3.1 Număr de ore pe săptămână	3	din care:	3.2 Curs	2	3.3 Seminar	-	3.3 Laborator	1	3.3 Proiect	-
3.4 Număr de ore pe semestru	42	din care:	3.5 Curs	28	3.6 Seminar	0	3.6 Laborator	14	3.6 Proiect	-
3.7 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										21
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										30
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										20
(d) Tutoriat										10
(e) Examinări										2
(f) Alte activități:										-
3.8 Total ore studiu individual (suma (3.7(a))...3.7(f)))					83					
3.9 Total ore pe semestru (3.4+3.8)					125					
3.10 Numărul de credite					5					

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Programare, Inginerie Software
4.2 de competențe	Competențele disciplinelor de mai sus

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Tabla, proiector, calculator
5.2. de desfășurare a seminarului / laboratorului / proiectului	Calculatoare, software specific

6. Competențele specifice acumulate

Competențe profesionale	• analizeaza specificatii software • aliniaza software-ul la arhitecturile de sistem • creeaza modele de date • creeaza softuri • defineste cerinte tehnice • defineste procesul • proiecteaza sistemul informatic • rezolva probleme ale sistemelor TIC • supravegheaza dezvoltarea de software • utilizeaza biblioteci de software • utilizeaza sabloane de proiectare de software
Competențe transversale	Absolventul: • dezvoltă o abordare analitică • adoptă o abordare proactivă • dezvoltă strategii pentru rezolvarea problemelor • este deschis la schimbare

7. Rezultatele așteptate ale învățării

Cunoștințe	Studentul cunoaște: • ciclul de viață al dezvoltării sistemelor • instrumente pentru gestionarea configurației software • limbajul unificat de modelare (UML) • modelarea orientată pe obiecte • modelarea proceselor de afaceri • proiectarea sistemelor • instrumente pentru gestionarea configurației software • bune practici pentru sistemul de backup • cadre software (framework-uri) • ingineria calculatoarelor
Aptitudini	Studentul este capabil să: • adune feedbackul clienților față de aplicații • alinieze software-ul la arhitecturile de sistem • analizeze cerințe de afaceri • analizeze specificații software • creeze diagrame de proces • creeze aplicații software • definească arhitectura software • definească cerințe tehnice • definească procese • dezvolte prototipuri software • efectueze analiza riscurilor
Responsabilitate și autonomie:	Studentul are capacitatea de a lucra independent pentru: • a dezvolta o abordare analitică • a adopta o abordare proactivă • a dezvolta strategii pentru rezolvarea problemelor • a fi deschis la schimbare

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

8.1 Obiectivul general al disciplinei	Cunoașterea și înțelegerea domeniului de fiabilitate software ca parte a calității software
---------------------------------------	---

8.2 Obiectivele specifice	▪ Studiul unor modele de fiabilitate software ▪ Dobandirea capacitatii de a produce estimări pentru fiabilitatea unui sistem software ▪ Intelegerea principalelor dificultati in atingerea unor nivele foarte ridicate de fiabilitate ▪ Estimarea și predicția fiabilității unui sistem software ▪ Dobandirea unor abilitati teoretice si practice de testare a sistemelor software
---------------------------	---

9. Conținuturi

9.1 Curs

9.1 Curs	Nr. ore	Metode de predare	Observații
Introducere	2	Expunere la tablă, prezentare cu videoproiectorul, discuții interactive.	
Prevenirea defectelor software	2		
Detectarea si eliminarea defectelor (1)	2		
Detectarea si eliminarea defectelor (2)	2		
Detectarea si eliminarea defectelor (3)	2		
Toleranta la erori si defecte	2		
Tehnici de creștere a fiabilității software	2		
Modelarea fiabilității software	2		
Tehnici de analiza predicției	2		
Analiza fiabilității software pe bază de măsurători	2		
Modele de fiabilitate software	2		
Simularea fiabilității software	2		
Studii de caz.	2		
Studii de caz.	2		
Bibliografie: 1. I. Sommerville, <i>Software Engineering (7th edition)</i> , Pearson Education Ltd., England, 2004. ISBN. [BCATCALC]			
9.2 Seminar / laborator / proiect	Nr. ore	Metode de predare	Observații
Clasificarea ortogonală a defectelor.	2	Lucrari practice folosind unelte software specifice.	
Recovery blocks.	2		
N-version Programming.	2		
N self-checking Programming.	2		
Profil operațional.	2		
Analiza fiabilității software pe bază de măsurători.	2		
Unelte de fiabilitate software.	2		
Bibliografie: I. Sommerville, <i>Software Engineering (7th edition)</i> , Pearson Education Ltd., England, 2004. ISBN. [BCATCALC]			

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

Disciplina corespunde asteptarilor comunitatii epistemice si a angajatorilor din domeniu, in special in ceea ce priveste activitatea in domeniul ingineriei software si a asigurarii calitatii sistemelor software, dar nu numai.

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare (și forma evaluare: continuă/sumativă)	11.3 Pondere din nota finală
11.4 Curs	Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de curs	Examen scris (E) (Evaluare sumativa)	70%

11.5 Seminar/Laborator /Proiect	Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de aplicații	Teme de laborator (L) - platforma MS Teams (Evaluare continuă)	30%
11.6 Standard minim de performanță: $E \geq 50\%$; $L \geq 50\%$; Nota finală disciplină: $N = 0.7 * E + 0.3 * L$			

Data completării: 29.04.2026	Titulari	Titlu Prenume NUME	Semnătura
	Curs	Prof.dr.ing. Alin SUCIU	
	Aplicații	Prof.dr.ing. Alin SUCIU	

Data avizării în Consiliul Departamentului Calculatoare	Director Departament, Prof.dr.ing. Rodica Potolea
Data aprobării în Consiliul Facultății de Automatică și Calculatoare	Decan, Prof.dr.ing. Vlad Mureșan