

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
1.8 Codul disciplinei	14.

2. Date despre disciplină

2.1 Denumirea disciplinei	Fiabilitatea programelor				
2.2 Titularii de curs	Prof. dr. ing. Alin Suciu – alin.suciu@cs.utcluj.ro				
2.3 Titularul/Titularii activităților de seminar/laborator/proiect	Prof. dr. ing. Alin Suciu – alin.suciu@cs.utcluj.ro				
2.4 Anul de studiu	II	2.5 Semestrul	3	2.6 Tipul de evaluare (E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DA – de aprofundare, DS – de sinteza, DC – complementară				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Timpul total estimat

3.1 Număr de ore pe săptămână	3	din care:	Curs	2	Seminar	-	Laborator	1	Proiect	-
3.2 Număr de ore pe semestru	42	din care:	Curs	28	Seminar	-	Laborator	14	Proiect	-
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										20
(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										3
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))							83			
3.5 Total ore pe semestru (3.2+3.4)							125			
3.6 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, platforme online
5.2. de desfășurare a seminarului / laboratorului / proiectului	Calculatoare, software specific, platforme online

6. Competențele specifice acumulate

6.1 Competențe profesionale	<p>C4 - Integrarea contextuală și integritatea sistemelor software complexe</p> <ul style="list-style-type: none"> • C4.1 - Demonstrarea cunoașterii și înțelegerii elementelor de interoperabilitate și integrare specifice sistemelor software, luate atât în ansamblu cât și pe module • C4.2 - Folosirea unor cunoștințe interdisciplinare pentru adaptarea sistemelor software complexe în raport cu cerințele dinamice ale domeniului de aplicații
-----------------------------	---

	<ul style="list-style-type: none"> • C4.3 - Utilizarea combinată a unor principii și metode clasice și originale pentru integrarea componentelor unor sisteme de calcul complexe • C4.4 - Folosirea standardelor de calitate, siguranță și securitate în prelucrarea informațiilor și în integrarea sistemelor software complexe. <p>C4.5 - Realizarea de proiecte interdisciplinare, incluzând identificarea și analiza problemei, elaborarea specificațiilor, proiectarea software, implementarea testarea funcțională și evaluarea criteriilor de calitate, securitate și de performanță specifice, precum și validarea sistemului software integrat.</p> <p>C5 - Îmbinarea creativă a cunoștințelor multidisciplinare din domeniul științei calculatoarelor și tehnologiei informației în vederea cercetării, specificării, proiectării, optimizării, implementării, testării și evaluării de teorii, algoritmi, tehnici, metode și metodologii originale specifice sistemelor software complexe.</p> <ul style="list-style-type: none"> • C5.1 - Demonstrarea cunoașterii metodologiei de cercetare, proiectare, implementare, optimizare și testare a sistemelor de calcul complexe • C5.2 - Demonstrarea capacității de a analiza și interpreta situații noi prin prisma cunoștințelor fundamentale din domeniul calculatoarelor și tehnologiei informației <p>C5.3 - Îmbinarea creativă, bazată pe descoperirea de legături semantice și funcționale noi, a diferite principii de proiectare moderne din domeniul calculatoarelor și tehnologiei informației pentru rezolvarea unor probleme de optimizare</p> <ul style="list-style-type: none"> • C5.4 - Fundamentarea activității de cercetare și proiectare inovativă din domeniul calculatoarelor pe criterii corecte de evaluare • C5.5 - Realizarea de activități de cercetare cu finalitate practică demonstrată prin prototipuri software și / sau hardware funcționale.
6.2 Competențe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Cunoașterea și înțelegerea domeniului de fiabilitate software ca parte a calității software
7.2 Obiectivele specifice	<ul style="list-style-type: none"> ▪ Studiul unor modele de fiabilitate software ▪ Dobandirea capacității de a produce estimări pentru fiabilitatea unui sistem software ▪ Înțelegerea principalelor dificultăți în atingerea unor nivele foarte ridicate de fiabilitate ▪ Estimarea și predicția fiabilității unui sistem software ▪ Dobandirea unor abilități teoretice și practice de testare a sistemelor software

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Introducere	2	Expunere la tablă, prezentare cu videoproiectorul, discuții interactive. Platforme online	Nu sunt
Prevenirea defectelor software	2		
Detectarea și eliminarea defectelor (1)	2		
Detectarea și eliminarea defectelor (2)	2		
Detectarea și eliminarea defectelor (3)	2		
Toleranța la erori și 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 (<i>bibliografia minimală a disciplinei conținând cel puțin o lucrare bibliografică de referință a disciplinei, care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)			
1. M. R. Lyu, <i>Handbook of Software Reliability Engineering</i> , McGraw Hill Publ. Company, New York, 1996, ISBN 0-07-039400-8 (disponibilă on-line).			
2. M. R. Lyu, <i>Software Fault Tolerance</i> , John Wiley & Sons Ltd., New York, 1995, ISBN 0-471-95058-8 (disponibilă on-line).			
3. I. Sommerville, <i>Software Engineering (7th edition)</i> , Pearson Education Ltd., England, 2004. ISBN. [BCATCALC]			
4. R. S. Pressman, <i>Software Engineering – A Practitioner’s Approach (5th edition)</i> , McGraw Hill Publ. Company, UK, 2000. ISBN 0-07-709677-0 [BCATCALC]			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
Clasificarea ortogonală a defectelor.	2	Lucrari practice folosind unelte software specifice si platforme online	Nu sunt
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>bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)			
1. M. R. Lyu, <i>Handbook of Software Reliability Engineering</i> , McGraw Hill Publ. Company, New York, 1996, ISBN 0-07-039400-8 (disponibilă on-line).			
2. M. R. Lyu, <i>Software Fault Tolerance</i> , John Wiley & Sons Ltd., New York, 1995, ISBN 0-471-95058-8 (disponibilă on-line).			
3. I. Sommerville, <i>Software Engineering (7th edition)</i> , Pearson Education Ltd., England, 2004. ISBN. [BCATCALC]			
4. R. S. Pressman, <i>Software Engineering – A Practitioner’s Approach (5th edition)</i> , McGraw Hill Publ. Company, UK, 2000. ISBN 0-07-709677-0 [BCATCALC]			

*Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

Disciplina corespunde așteptărilor comunității epistemice și a angajatorilor din domeniu, în special în ceea ce privește activitatea în domeniul ingineriei software și a asigurării calității sistemelor software, dar nu numai.

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de curs	Examen scris și/sau Examen oral online sau onsite (E)	50%
Seminar			
Laborator	Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de aplicații	Teme de laborator online sau onsite (L)	50%
Proiect			
Standard minim de performanță: E ≥ 50% ; L ≥ 50%			
Nota finală disciplină: N = 0.5*L + 0.5*E			

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
23.05.2023	Curs	Prof. dr. ing. Alin Suci	
	Aplicații	Prof. dr. ing. Alin Suci	

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. Liviu Miclea