

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Facultatea de Automatică și Calculatoare
1.3 Departamentul	Automatică
1.4 Domeniul de studii	Ingineria Sistemelor
1.5 Ciclul de studii	Master
1.6 Programul de studii / Calificarea	Informatică Aplicată în Ingineria Sistemelor Complexe
1.7 Forma de învățământ	IF – învățământ cu frecvență

2. Date despre disciplină

2.1 Denumirea disciplinei	Asigurarea calitatii programelor			Codul disciplinei	10.00
2.2 Titularul de curs	Prof. dr. ing. Liviu Miclea – Liviu.Miclea@aut.utcluj.ro				
2.3 Titularul activităților de seminar / laborator / proiect	S.I.dr.ing. Iulia Stefan – Iulia.Stefan@aut.utcluj.ro				
2.4 Anul de studiu	1	2.5 Semestrul	2	2.6 Tipul de evaluare	E
2.7 Regimul disciplinei	Categoria formativă				DA
	Opționalitate				DI

3. Timpul total estimate

3.1 Număr de ore pe săptămână	3	din care:	3.2 Curs	2	3.3 Seminar	0	3.3 Laborator	1	3.3 Proiect	0
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	0
3.7 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										15
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										14
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										23
(d) Tutoriat										3
(e) Examinări										3
(f) Alte activități:										0
3.8 Total ore studiu individual (suma (3.7(a)...3.7(f)))					58					
3.9 Total ore pe semestru (3.4+3.8)					100					
3.10 Numărul de credite					4					

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	N/A
4.2 de competențe	Matematica (algebra, logica, teoria grafurilor) si statistica Cunostinte de programare in limbaje de nivel inalt. Cunostinte de ingineria programarii

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	
5.2. de desfășurare a seminarului / laboratorului / proiectului	Prezența la laborator este obligatorie

6. Competențele specifice acumulate

Competențe profesionale	• Efectuează cercetare științifică • Prezintă rezultatele analizelor • Realizează planificarea de resurse • Stabilește obiective de asigurare a calității • Efectuează analiza riscurilor • Furnizează documentație tehnică • Asigură managementul de proiect • Sintetizează informații • Gestionează date în domeniul cercetării • Gândește în mod abstract • Interacționează profesional în mediile de cercetare profesionale • Dezvoltă aplicații de procesare de date • Aplică tehnici de analiză statistică • Desfășoară revizuirea codului în TIC • Definește arhitectura software • Utilizează biblioteci software • Analizează specificații software
Competențe transversale	• Dă dovadă de inițiativă • Gândește analitic • Aplică cunoștințe științifice, tehnologice și ingineresti

7. Rezultatele așteptate ale învățării

Cunoștințe	Studentul va cunoaște: • conceptele fundamentale și metodologia ingineriei sistemelor aplicate în dezvoltarea aplicațiilor informatice și a sistemelor complexe de control • metodele de simulare, modelare, evaluare și optimizare a performanțelor aplicațiilor software și sistemelor de control • principiile arhitecturii software, instrumentele și mediile de dezvoltare pentru aplicații informatice și sisteme autonome • standardele de calitate, securitate și siguranță, precum și codurile de etică și integritate academică aplicate în cercetare și dezvoltare • principiile managementului de proiect, planificarea resurselor, analiza riscurilor și evaluarea fezabilității proiectelor • tehnici de procesare a datelor, analiză statistică și diseminare a rezultatelor în mediul științific
Abilități	Studentul va fi capabil: • să proiecteze, dezvolte și testeze aplicații informatice și sisteme complexe de control utilizând metode și tehnologii moderne • să aplice metode de analiză, simulare și modelare pentru optimizarea soluțiilor tehnice și informatice • să realizeze prototipuri software și/sau hardware funcționale pentru demonstrarea rezultatelor cercetării • să comunice profesional, să colaboreze în echipă și să disemineze rezultatele cercetării în comunități științifice • să folosească instrumente de management al proiectelor pentru planificarea, monitorizarea și evaluarea activităților • să identifice probleme și să propună soluții inovative în domeniul ingineriei sistemelor, TIC și aplicațiilor industriale

Responsabilitate și autonomie:	Studentul: • Va fi responsabil pentru gestionarea corectă a resurselor de proiect, planificarea activităților și evaluarea riscurilor asociate • Va gestiona autonom proiecte de inginerie software, industrială și sisteme incorporate • Va fi capabil să își asume responsabilitatea pentru calitatea soluțiilor implementate • Va gestiona proiecte profesionale sau de cercetare • Va fi capabil să lucreze eficient în echipe
--------------------------------	---

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

8.1 Obiectivul general al disciplinei	Principalele tehnici de testare a programelor: testare funcțională (black-box), testare structurală (white-box), testare statistică bazată pe utilizare. Modele de bază utilizate în testarea programelor. Activitățile, managementul, automatizarea testării și aspecte asociate ca sub-fazele testării, organizarea echipelor, procesul de testare, rolul și responsabilitățile angajaților, utilitate de automatizare a testării
8.2 Obiectivele specifice	Înșușirea unor metode eficiente de dezvoltare de secvențe de test a programelor. Cunoașterea unor metodologii de calcul a fiabilității programelor. Utilizarea unor medii pentru crearea și dezvoltarea de teste. Abilitatea de a coordona procesul de testare operativă și eficiența a programelor.

9. Conținuturi

9.1 Curs	Nr. ore	Metode de predare	Observații
1. Calitatea programelor: Privire generala si fundamente	2	Expunerea didactică, conversația didactică, problematizarea	
2. Asigurarea calitatii (AC): fundamente, AC in contextul dezvoltarii programelor	2		
3. Testarea programelor - Concepte, probleme si tehnici: Testare functionala (black box) vs. structurala (white box), Testare statistica bazata pe utilizare (UBS – Usage-Based Statistical testing) vs. Testare bazata pe acoperirea defectelor (CB – Coverage-Based testing)	2		
4. Activitatile, managementul si automatizarea testarii programelor	2		
5. Testare CB si UBS utilizand liste de verificare si partitii	2		
6. Partitionarea domeniului de intrare si testarea marginala	2		
7. Testare CB si UBS utilizand masini cu stari finite (FSM) si lanturi Markov	2		
8. Testarea fluxului de control, dependentei datelor si a interactiunilor	2		
9. Tehnicile de testare: adaptare, specializare si integrare	2		
10. Generarea testelor din modele de mașini cu stări finite (FSM)	2		
11. Asigurarea calitatii (QA – Quality Assurance) dincolo de testare: prevenirea defectiunilor si imbunatatirea procesului	2		
12. Inspectia programelor	2		
13. Verificarea formala	2		
14. Toleranta la defecte si controlul defectelor	2		
Bibliografie			

1. Jeff Tian, Software Quality Engineering – Testing, Quality Assurance, and Quantifiable Improvement, John Wiley & Sons.
2. W.E. Perry, Effective Methods for Software Testing, Wiley Publishing.
3. M.B. Chrissis, M. Konrad, S. Shrun, CMMI Guidelines for Process Integration and Product Improvement, Addison-Wesley.
4. Glenford J. Meyers, The Art of Software Testing, second edition, John Wiley & Sons, Inc.
5. Michael R. Lyu, Editor in Chief, Handbook of Software Reliability Engineering, IEEE Computer Society Press, McGraw-Hill.
6. Marnie L. Hutcheson, Software Testing Fundamentals: Methods and Metrics, John Wiley & Sons, Inc.
7. Al. Balog (ed), Calitatea sistemelor interactive, Ed. Matrix Rom, Bucuresti.

9.2 Seminar / laborator / proiect	Nr. ore	Metode de predare	Observații
L1,2. Aplicații de generare a testelor “black-box”, bazate pe date	2	Studii de caz, prezentarea unor metode și secvențe de testare, programe, discuții	
L3,4. Aplicații de generare a testelor “white-box”, bazate pe structură	2		
L5,6. Aplicații de generare a testelor “black-box”, bazate pe structură	2		
L7,8. Aplicații de generare a testelor interfetelor	2		
L9,10. Studii de caz: Testarea unor programe COTS (Commercial off-the-shelf) utilizând utilitare	2		
L11,12. Studii de caz: Testarea aplicațiilor web	2		
L13,14. Aplicații de calcul a fiabilității programelor	2		

Bibliografie

1. Jeff Tian, Software Quality Engineering – Testing, Quality Assurance, and Quantifiable Improvement, John Wiley & Sons.
2. W.E. Perry, Effective Methods for Software Testing, Wiley Publishing.
3. M.B. Chrissis, M. Konrad, S. Shrun, CMMI Guidelines for Process Integration and Product Improvement, Addison-Wesley.
4. Glenford J. Meyers, The Art of Software Testing, second edition, John Wiley & Sons, Inc.
5. Michael R. Lyu, Editor in Chief, Handbook of Software Reliability Engineering, IEEE Computer Society Press, McGraw-Hill.
6. Marnie L. Hutcheson, Software Testing Fundamentals: Methods and Metrics, John Wiley & Sons, Inc.
7. Al. Balog (ed), Calitatea sistemelor interactive, Ed. Matrix Rom, Bucuresti.

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

Subiectele de la laborator sunt inspirate din aplicațiile unor firme din Cluj-Napoca, precum Siemens, Arobs, Emerson etc.

Temele de la proiect corespund unor aplicații ale firmelor de la noi din țară.

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	Cunoștințe teoretice și un proiect de curs	Examen oral, bazat întrebări teoretice (T, 20%) și un proiect de curs (PC, 30%)	60% (10% din oficiu)
11.5 Seminar/Laborator /Proiect	Teme de casa. Elaborarea unui proiect de sinteză.	Evaluare laborator (PL) bazat pe proiectul de sinteză, evaluat oral	40%

11.6 Standard minim de performanțăStandard minim de performanță: $T \geq 5$ și $PL \geq 5$ și $PC \geq 5$; $N = 1 + 0.4 \cdot PL + 0.3 \cdot PE + 0.2 \cdot T$.

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
01.09.2025	Curs	Prof.dr.ing. Liviu Miclea	
	Aplicații	Ș.l.dr.ing. Iulia Ștefan	

Data avizării în Consiliul Departamentului de Automatică

Director Departament Automatică

Prof.dr.ing. Honoriu Vălean

Data aprobării în Consiliul Facultății de Automatică și
Calculatoare

Decan

Prof.dr.ing. Vlad Mureșan
