

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
1.8 Codul disciplinei	10

2. Date despre disciplină

2.1 Denumirea disciplinei	Asigurarea calitatii programelor				
2.2 Titularul de curs	Prof. dr. ing. Liviu Miclea – Liviu.Miclea@aut.utcluj.ro				
2.3 Titularul/Titularii activităților de seminar/laborator/proiect	S.l.dr.ing. Iulia Stefan – Iulia.Stefan@aut.utcluj.ro				
2.4 Anul de studiu	1	2.5 Semestrul	1	2.6 Tipul de evaluare (E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DA – de aprofundare, DS – de sinteză, DC – complementară				DA
	DI – impusă, DO – opțională, DFac – facultativă				DI

3. Timpul total estimat

3.1 Număr de ore pe săptămână	3	din care:	Curs	2	Seminar	0	Laborator	1	Proiect	0
3.2 Număr de ore pe semestru	42	din care:	Curs	28	Seminar	0	Laborator	14	Proiect	0
3.3 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
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))	58									
3.5 Total ore pe semestru (3.2+3.4)	100									
3.6 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) și statistica Cunostinte de programare în limbaje de nivel înalt. 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

6.1 Competențe profesionale	C4. Integrarea contextuală și integritatea sistemelor complexe de control și ale rețelelor industriale. C5. Îmbinarea creativă a cunoștințelor multidisciplinare din domeniul ingineriei sistemelor, calculatoarelor și tehnologiei informației în vederea cercetării, proiectării, optimizării, implementării și testării de teorii, algoritmi și metode originale specifice sistemelor complexe de control, și a rețelelor industriale.
-----------------------------	--

6.2 Competențe transversale	N/A
-----------------------------	-----

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Principalele tehnici de testare a programelor: testare functională (black-box), testare structurală (white-box), testare statistică bazată pe utilizare. Modele de baza 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
7.2 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.

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
1. Calitatea programelor: Privire generală și fundamente	2	Expunerea didactică, conversația didactică, problematizarea	
2. Asigurarea calității (AC): fundamente, AC în contextul dezvoltării programelor	2		
3. Testarea programelor - Concepte, probleme și tehnici: Testare funcțională (black box) vs. structurală (white box), Testare statistică bazată pe utilizare (UBS – Usage-Based Statistical testing) vs. Testare bazată pe acoperirea defectelor (CB – Coverage-Based testing)	2		
4. Activitățile, managementul și automatizarea testării programelor	2		
5. Testare CB și UBS utilizând liste de verificare și partiții	2		
6. Partitionarea domeniului de intrare și testarea marginală	2		
7. Testare CB și UBS utilizând mașini cu stări finite (FSM) și lanțuri Markov	2		
8. Testarea fluxului de control, dependenței datelor și a interacțiunilor	2		
9. Tehnicile de testare: adaptare, specializare și integrare	2		
10. Generarea testelor din modele de mașini cu stări finite (FSM)	2		
11. P Asigurarea calității (QA – Quality Assurance) dincolo de testare: prevenirea defectiunilor și îmbunătățirea procesului	2		
12. Inspectia programelor	2		
13. Verificarea formală	2		
14. Toleranța la defecte și controlul erorilor	2		

Bibliografie (*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*)

1. Jeff Tian, Software Quality Engineering – Testing, Quality Assurance, and Quantifiable Improvement, John Wiley & Sons, 2005
2. W.E. Perry, Effective Methods for Software Testing, Wiley Publishing, 2006
3. M.B. Chrissis, M. Konrad, S. Shrun, CMMI Guidelines for Process Integration and Product Improvement, Addison-Wesley, 2003
4. Glenford J. Meyers, The Art of Software Testing, second edition, John Wiley & Sons, Inc., 2004
5. Michael R. Lyu, Editor in Chief, Handbook of Software Reliability Engineering, IEEE Computer Society Press, McGraw-Hill, 1996
6. Marnie L. Hutcheson, Software Testing Fundamentals: Methods and Metrics, John Wiley & Sons, Inc., 2003
7. Al. Balog (ed), Calitatea sistemelor interactive, Ed. Matrix Rom, București, 2004

8.2 Aplicații (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	
L3,4. Aplicații de generare a testelor “white-box”, bazate pe structură	2		

L5,6. Aplicații de generare a testelor “black-box”, bazată pe structura	2	de testare programe, discuții	
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 (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)

1. Jeff Tian, Software Quality Engineering – Testing, Quality Assurance, and Quantifiable Improvement, John Wiley & Sons, 2005
2. W.E. Perry, Effective Methods for Software Testing, Wiley Publishing, 2006
3. M.B. Chrissis, M. Konrad, S. Shrun, CMMI Guidelines for Process Integration and Product Improvement, Addison-Wesley, 2003
4. Glenford J. Meyers, The Art of Software Testing, second edition, John Wiley & Sons, Inc., 2004
5. Michael R. Lyu, Editor in Chief, Handbook of Software Reliability Engineering, IEEE Computer Society Press, McGraw-Hill, 1996
6. Marnie L. Hutcheson, Software Testing Fundamentals: Methods and Metrics, John Wiley & Sons, Inc., 2003
7. Al. Balog (ed), Calitatea sistemelor interactive, Ed. Matrix Rom, Bucuresti, 2004

*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

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

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

10. Evaluare

On-site

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Abilitatea rezolvării problemelor	Examen oral, bazat pe un proiect de curs (PC) și întrebări din teorie (T)	60% (10% din oficiu)
Seminar			
Laborator	Teme de casa. Elaborarea unui proiect de sinteză.	Evaluare laborator (PL) bazat pe proiectul de sinteză, evaluat oral	40%
Proiect			

Standard minim de performanță: $PL \geq 5$, $PC \geq 5$;
 $N = 1 + 0.4 * PL + 0.3 * PE + 0.2 * T$.

On-line

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Abilitatea rezolvării problemelor	Examen oral, on line, cu teams, bazat pe un proiect de curs (PC) și întrebări din teorie (T)	60% (10% din oficiu)
Seminar			

Laborator	Teme de casa. Elaborarea unui proiect de sinteză.	Evaluare laborator (PL) bazat pe proiectul de sinteza, evaluat on-line, cu teams	40%
Proiect			
Standard minim de performanță: $PL \geq 5$, $PE > 5$; $N = 1 + 0.4 * PL + 0.3 * PE + 0.2 * T$.			

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
20.03.2023	Curs	Prof.dr.ing. Liviu Miclea	
	Aplicații	S.I.dr.ing. Iulia Stefan	

Data avizării în Consiliul Departamentului de Automatica	Director Departament Automatica. Prof.dr.ing. Honoriu Vamean

Data aprobării în Consiliul Facultății de Automatica și Calculatoare	Decan Prof.dr.ing. Liviu Miclea
