

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
1.7 Forma de învățământ	IF – învățământ cu frecvență
1.8 Codul disciplinei	4

2. Date despre disciplină

2.1 Denumirea disciplinei	Testarea aplicațiilor software				
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. Dan Gota – Dan.Gota@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										23
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										22
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										31
(d) Tutoriat										4
(e) Examinări										3
(f) Alte activități:										0
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	• N/A
4.2 de competențe	• Notiuni de programarea calculatoarelor

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	
5.2. de desfășurare a seminarului / laboratorului / proiectului	

6. Competențele specifice acumulate

6.1 Competențe profesionale	<ul style="list-style-type: none"> • C2 - Proiectarea, implementarea, testarea și mentenanța aplicațiilor și a bazelor de date <ul style="list-style-type: none"> ○ C2.1 - Demonstrarea cunoașterii tehnologiilor, mediilor de programare și conceptelor specifice informaticii aplicate ○ C2.2 - Analiza și explicarea rolului, interacțiunilor și al modului de funcționare al componentelor software dezvoltate pe baza celor mai noi metodologii de proiectare propuse în literatura științifică pentru informatica aplicata ○ C2.3 - Analizarea în mod critic și descoperirea aspectelor susceptibile de optimizare, urmată de aplicarea unor tipare de soluții inovative
-----------------------------	---

	<p>adevate pentru aplicații informatice complexe capabile să răspundă unor cerințe moderne</p> <ul style="list-style-type: none"> ○ C2.4 - Evaluarea comparativă, sintetică, inclusiv experimentală, a alternativelor de rezolvare pentru optimizarea performanțelor, pe baza criteriilor de utilizabilitate ○ C2.5 - Dezvoltarea și implementarea de soluții informatice originale pentru problemele specifice domeniului, pornind de la un set de cerințe informal specificate • C5 - Folosirea de cunoștințe avansate referitoare la managementul de proiect, în dezvoltarea și mentenanța aplicațiilor informatice <ul style="list-style-type: none"> ○ C5.1 - Identificarea metodelor de management de proiect utilizate în dezvoltarea și mentenanța aplicațiilor informatice. ○ C5.2 - Utilizarea capacității de a analiza și interpreta situații noi prin prisma cunoștințelor multidisciplinare din domeniul managementului de proiecte informatice ○ C5.3 - Îmbinarea creativă, bazată pe descoperirea de legături semantice și funcționale noi, a diferite principii de proiectare moderne din domeniul managementului de proiect pentru rezolvarea unor probleme de informatica aplicată ○ C5.4 - Evaluarea metodelor de management al proiectelor destinate sistemelor informatice aplicate, a gradului de realizabilitate, a fezabilității acestora, fundamentarea unor soluții de implementare, evaluarea riscurilor. ○ C5.5 - Organizarea și conducerea de activități specifice de proiectare, elaborarea conceptuală de proiecte de cercetare cu grad de complexitate redus.
6.2 Competențe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Obiectivul principal al acestei discipline este însușirea de către studenți a conceptelor de bază ale teoriei testării programelor, a tehnicilor de testare la diferite nivele (unitate, integrare, sistem și acceptanță), a tehnicilor de proiectare, planificare și execuție a testelor precum și însușirea unor noțiuni de fiabilitate a programelor
7.2 Obiectivele specifice	<p>În cadrul acestei discipline studenții vor învăța:</p> <ul style="list-style-type: none"> • practici care susțin producția unui software de calitate • să dezvolte teste pentru sisteme complexe • să utilizeze aplicații software de testare la diferite nivele. • să construiască echipe de testare

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
1. Concepte de baza: calitatea programelor; defect, eroare, esec; secvența de test	2	Expunerea didactică, conversația didactică, problematizarea	
2. Teoria testării programelor	2		
3. Testarea la nivel de unitate	2		
4. Testarea fluxului de control	2		
5. Testarea fluxului de date	2		
6. Testarea domeniului	2		
7. Testarea de integrare a sistemului	2		
8. Categoriile de testare a sistemului	2		
9. Testare funcțională	2		
10. Generarea testelor din modele de mașini cu stări finite (FSM)	2		
11. Proiectarea testului de sistem. Planificarea și	2		

automatizarea testului de sistem, Execuția testului de sistem			
12. Testarea de acceptanta	2		
13. Fiabilitatea programelor	2		
14. Organizarea echipei de test	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. Kshirasagar Naik, Priyadarshi Tripathy, Software Testing and Quality Assurance. Theory and Practice, John Wiley & Soons, Inc., 2008			
2. Jeff Tian, Software Quality Engineering – Testing, Quality Assurance, and Quantifiable Improvement, John Wiley & Sons, 2005			
3. W.E. Perry, Effective Methods for Software Testing, Wiley Publishing, 2006			
4. M.B. Chrissis, M. Konrad, S. Shrun, CMMI Guidelines for Process Integration and Product Improvement, Addison-Wesley, 2003			
5. Al. Balog (ed), Calitatea sistemelor interactive, Ed. Matrix Rom, Bucuresti, 2004			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
L1,2. Aplicații de testare la nivel de unitate	2	Studii de caz, prezentarea unor metode si secvențe de testare programe, discuții	
L3,4. Aplicații de testare a fluxului de control	2		
L5,6. Aplicații de testare a fluxului de date	2		
L7,8. Aplicații de testare a domeniului	2		
L9,10. Aplicații de testare de integrare a sistemului	2		
L11,12. Aplicații de testare funcționala	2		
L13,14. Aplicații de calcul a fiabilității programelor	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>)			
6. 1. Kshirasagar Naik, Priyadarshi Tripathy, Software Testing and Quality Assurance. Theory and Practice, John Wiley & Soons, Inc., 2008			
7. Jeff Tian, Software Quality Engineering – Testing, Quality Assurance, and Quantifiable Improvement, John Wiley & Sons, 2005			
8. W.E. Perry, Effective Methods for Software Testing, Wiley Publishing, 2006			
9. M.B. Chrissis, M. Konrad, S. Shrun, CMMI Guidelines for Process Integration and Product Improvement, Addison-Wesley, 2003			
10. 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

Disciplina este fundamentală pentru o carieră în informatică și programarea calculatoarelor. Conținutul îmbină cunoștințe teoretice cu aplicații și se concentrează pe formularea și rezolvarea unor probleme specifice de testare a aplicațiilor software care pot apare într-o diversitate de domenii din inginerie.

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) si intrebari din teorie (T)	60% (10% din oficiu)
Seminar			
Laborator	Abilitatea rezolvării problemelor. Prezență.	Evaluare laborator (PL) bazat pe	40%

	Interactivitate	proiectul de sinteza, evaluat oral	
Proiect			
S Standard minim de performanță: PL>=5, PE>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) si intrebari din teorie (T)	60% (10% din oficiu)
Seminar			
Laborator	Abilitatea rezolvării problemelor. Prezență. Interactivitate	Evaluare laborator (PL) bazat pe proiectul de sinteza, evaluat on-line, cu zoom	40%
Proiect			
Standard minim de performanță: PL>=5, PE>5; N=1+0.4*PL+0.3*PE+0.2*T.			

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
27.06.2022	Curs	Prof.dr.ing. Liviu Miclea	
	Aplicații	S.I.dr.ing. Dan Gota	

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

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