

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 Ciclu 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	Ingineria Programării				Codul disciplinei	13.00
2.2 Titularul de curs	Prof. dr. ing. Dînsoreanu Mihaela - Mihaela.Dinsoreanu@cs.utcluj.ro					
2.3 Titularul activităților de seminar / laborator / proiect	Prof. dr. ing. Dînsoreanu Mihaela - Mihaela.Dinsoreanu@cs.utcluj.ro					
2.4 Anul de studiu	II	2.5 Semestrul	1	2.6 Tipul de evaluare	E	
2.7 Regimul disciplinei	Categoriza formativă					DA
	Opționalitate					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	14	Laborator	-	Proiect	-
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										30
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										20
(c) Pregătire seminar / laboratoare, teme, referate, portofolii și eseuri										25
(d) Tutoriat										5
(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	Proiectare software
4.2 de competențe	Elaborarea de tehnici, tehnologii, metode și metodologii specifice sistemelor informatice; Proiectarea inovativă a sistemelor informatice dedicate

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Proiector, tabla, calculator cu acces la internet
5.2. de desfășurare a seminarului / laboratorului / proiectului	Proiector, calculatoare cu acces la internet

6. Competențele specifice acumulate

6.1 Competențe profesionale	• analizează specificații software • aliniază software-ul la arhitecturile de sistem • creează modele de date • creează softuri • definește cerințe tehnice • definește procesul • definește arhitectura software • efectuează cercetare științifică • efectuează modificarea și transferul aplicațiilor în cloud • furnizează documentație tehnică • gestionează date în cloud și stocarea acestora • identifică cerințele clienților • identifică nevoile utilizatorilor de TIC • interacționează cu utilizatorii pentru a le afla cerințele • interpretează cerințe tehnice • interpretează texte tehnice • implementează resurse cloud • planifică migrarea către infrastructura cloud • proiectează arhitecturi de tip cloud • proiectează baza de date în cloud • proiectează interfața cu utilizatorul • proiectează sistemul informatic • rezolvă probleme ale sistemelor TIC • supraveghează dezvoltarea de software • utilizează biblioteci de software • utilizează șabloane de proiectare de software
6.2 Competențe transversale	Absolventul: • dezvoltă o abordare analitică • adoptă o abordare proactivă • dezvoltă strategii pentru rezolvarea problemelor • este deschis la schimbare • coordonează echipele tehnice

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 • formularea algoritmului unui proces • ingineria de securitate • legislația în domeniul securității TIC • managementul bazat pe procese • metodologiile de management de proiect în TIC • instrumente pentru gestionarea configurației software • cadre software (framework-uri) • ingineria calculatoarelor • monitorizarea și raportarea în domeniul tehnologiilor de tip cloud • programarea de sisteme TIC • software pentru medii integrate de dezvoltare (IDE) • tehnologii cloud • metodologii de management de proiect în TIC • concepte privind ușurința de utilizare a aplicațiilor (usability)
------------	--

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 • furnizeze rapoarte de analiză cost-beneficiu • supravegheze dezvoltarea de software • aplice teoria sistemelor de TIC • dezvolte prototipuri software • efectueze analiza riscurilor • efectueze modificarea și migrarea aplicațiilor în cloud • furnizeze consultanță în domeniul TIC • implementeze tehnici de stocare a datelor • proiecteze arhitecturi de tip cloud • proiecteze baze de date în cloud • proiecteze scheme ale bazelor de date • realizeze depanare TIC • realizeze planificarea resurselor • utilizeze instrumente de inginerie software asistată de calculator • utilizeze limbaje de interogare • utilizeze limbaje de marcare • conceapă complexitatea organizațională • dezvolte servicii de tip cloud computing • gestioneze datele în cloud și stocarea acestora • gestioneze planurile de recuperare în urma dezastrelor • implementeze sistemul de recuperare TIC • planifice migrarea către infrastructura cloud • proiecteze rețele de tip cloud • pună în aplicare măsuri de securitate și conformitate în domeniul tehnologiilor de tip cloud
Responsabilitate și autonomie:	Studentul are capacitatea de a lucra independent pentru: • a dezvolta o abordare analitică • a da dovadă de dorință de învățare • a dezvolta strategii pentru rezolvarea problemelor • a fi deschis la schimbare • a lucra eficient în echipe

8. Obiectivele disciplinei

8.1 Obiectivul general al disciplinei	Principalul obiectiv al acestei discipline este de a oferi informații specifice și de a pregăti studenții în vederea realizării de proiecte software conforme cerințele de calitate. Astfel, se urmărește conferirea capacității de a analiza diferite alternative arhitecturale și de proiectare, de a lua deciziile arhitecturale cele mai potrivite contextului în vederea proiectării oricărui tip de aplicație, cu accent pe optimizarea cât mai pronunțată a performanțelor acestuia.
---------------------------------------	--

8.2 Obiectivele specifice	Pentru atingerea acestor obiective generale, studenții vor: • Urmări să înțeleagă și să rezolve atât cerințele functionale ale unui sistem software cat si cerințele de calitate ale acestuia (disponibilitate, performanta, securitate, scalabilitate etc) • Studia solutii arhitecturale existente pe diferite nivele de granularitate (stiluri arhitecturale, sabloane arhitecturale si de proiectare) • Studia metrice de evaluare ale diferitelor aspecte calitative (complexitate, fiabilitate, disponibilitate etc.) • Învăța să analizeze cerințele si sa proiecteze alternative arhitecturale aplicabile; • Învăța să evalueze solutiile arhitecturale aplicand modele de evaluare specifice;
---------------------------	---

9. Conținuturi

9.1 Curs	Nr.ore	Metode de predare	Observații
Introducere	2	Expunere la tablă, prezentare cu videoproiectorul, discuții, materiale de curs, quizz-uri platforma Moodle	
Fundamente ale metricilor si masuratorilor	2		
Modele de proces si practici agile	2		
Ingineria cerintelor	2		
Planificare (masurarea dimensiunii, complexitatii, estimarea efortului)	2		
Planificare (estimarea timpului si costului)	2		
Monitorizare (metrici)	2		
Metrici ale defectelor	2		
Metrici de fiabilitate	2		
Metrici de disponibilitate	2		
Metrici de scalabilitate	2		
Securitate	2		
Modele de evaluare a arhitecturilor	2		
Recapitulare si concluzii	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. Software Metrics: A Rigorous and Practical Approach, by Norman Fenton, James Bieman, Third Edition, CRC Press, Inc., Boca Raton, FL, 2014 2. Software Engineering, 10th Edition, Ian Sommerville, Pearson ed. 2016 3. Righting software, Juval Lowy, O'Reilly, 2020 4. Software Architecture: The Hard Parts, Ford, Neal, Richards, Mark, Sadalage, Pramod, & Dehghani, Zhamak. (2021). O'Reilly Media, Inc. 5. Diferite articole			
9.2 Aplicații - seminar / laborator / proiect	Nr.ore	Metode de predare	Observații
Masurarea complexitatii	2	Expuneri la tablă, prezentari, discuții	
Estimarea efortului	2		
Planificarea proiectelor	2		
Metrici ale defectelor	2		
Metrici de fiabilitate	2		
Metrici de disponibilitate	2		
Metrici de scalabilitate	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. Software Metrics: A Rigorous and Practical Approach, by Norman Fenton, James Bieman, Third Edition, CRC Press, Inc., Boca Raton, FL, 2014
2. Software Engineering, 10th Edition, Ian Sommerville, Pearson ed. 2016
3. Righting software, Juval Lowy, O'Reilly, 2020
4. Software Architecture: The Hard Parts, Ford, Neal, Richards, Mark, Sadalage, Pramod, & Dehghani, Zhamak. (2021). O'Reilly Media, Inc.
5. Diferite articole

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

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

Întrucât această disciplină este foarte importantă pentru evaluarea calitativă a proiectării sistemelor software, conținutul ei este cât se poate de modern deoarece recapitulează principiile proiectării software, principiile proiectării proiectelor, apoi metrice de evaluare a proiectelor software din diferite perspective ale calitatii. Conținutul disciplinei a fost discutat cu actori importanți din acest domeniu, atât academici cât și industriali, din România, Europa și S.U.A. Disciplina a fost evaluată, o dată cu programul de studiu de master Tehnologia Informației în Economie, de către ARACIS.

11. 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	Quiz-uri pe parcurs, Examen scris - sumativ	60%
Seminar	Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de aplicații	Prezentarea unei topici de cercetare din domeniul cursului - sumativ	40%
Laborator	-	-	-
Proiect	-	-	-
Standard minim de performanță: nota seminar ≥ 5 , nota examen ≥ 5			

Data completării: 01.09.2025	Titulari	Titlu, prenume / NUME	Semnătura
	Curs	Prof.dr.ing. Mihaela DÎNȘOREANU	
	Aplicații	Prof.dr.ing. Mihaela DÎNȘOREANU	

Data avizării în Consiliul Departamentului Calculatoare
17.09.2025

Director Departament,
Prof.dr.ing. Rodica Potolea

Data aprobării în Consiliul Facultății de Automatică și Calculatoare
19.09.2025

Decan,
Prof.dr.ing. Vlad Muresan

