



FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea “Alexandru Ioan Cuza” din Iași
1.2 Facultatea	Facultatea de Informatică
1.3 Departamentul	Informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Master
1.6 Programul de studii / Calificarea	Ingineria sistemelor software

2. Date despre disciplină

2.1 Denumirea disciplinei	Calitatea sistemelor software						
2.2 Titularul activităților de curs	Lect. dr. Vlad Rădulescu						
2.3 Titularul activităților de seminar	Lect. dr. Vlad Rădulescu						
2.4 An de studiu	I	2.5 Semestru	2	2.6 Tip de evaluare	E	2.7 Regimul disciplinei	OP

* OB – Obligatoriu / OP – Opțional

3. Timpul total estimat (ore pe semestru și activități didactice)

3.1 Număr de ore pe săptămână	4	din care: 3.2 curs	2	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	56	din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp					ore
Studiu după manual, suport de curs, bibliografie și altele					40
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					30
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					40
Tutoriat					
Examinări					4
Alte activități					
3.7 Total ore studiu individual					110
3.8 Total ore pe semestru					170
3.9 Număr de credite					8

4. Precondiții (dacă este cazul)

4.1 De curriculum	-
4.2 De competențe	familiarizarea cu principiile ingineriei software

5. Condiții (dacă este cazul)

5.1 De desfășurare a cursului	-
5.2 De desfășurare a seminarului/ laboratorului	-



6. Competențe specifice acumulate

Competențe profesionale	C1. Identificarea de metodologii adecvate de dezvoltarea sistemelor software. C2. Identificarea modelelor și metodelor adecvate pentru rezolvarea unor probleme reale. C3. Realizarea unor proiecte informatice dedicate.
Competențe transversale	CT1. Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională.

7. Obiectivele disciplinei (din grila competențelor specifice acumulate)

7.1 Obiectivul general	Înțelegerea principalelor elemente care definesc calitatea sistemelor software. Familiarizarea cu metodele de testare și analiză a produselor software.
7.2 Obiectivele specifice	La finalizarea cu succes a acestei discipline, studenții vor fi capabili să: <ul style="list-style-type: none">▪ Descrie noțiunile principale legate de testarea software, analiza riscului, planificarea testării, măsurarea calității proiectelor software.▪ Utilizeze unelte de testare a software-ului.▪ Analizeze proiectele software din punct de vedere al riscului de apariție a defectelor.▪ Planifice testarea sistemelor software.▪ Decidă măsurile care se impun pentru îmbunătățirea procesului de dezvoltare a unui proiect software.

8. Conținut

8.1	Curs	Metode de predare	Observații (ore și referințe bibliografice)
1	Introducere	expunere, dezbateri, studii de caz, rezolvare de probleme	-
2	Testarea programelor	expunere, dezbateri, studii de caz, rezolvare de probleme	-
3	Defecte în sistemele software. Inspecția codului	expunere, dezbateri, studii de caz, rezolvare de probleme	-
4	Analiza riscului. Planuri de test	expunere, dezbateri, studii de caz, rezolvare de probleme	-
5	Nivele de testare: unit testing, integration testing, system testing, acceptance testing	expunere, dezbateri, studii de caz, rezolvare de probleme	-



6	Testare extremă. Testare regresivă	expunere, dezbateri, studii de caz, rezolvare de probleme	-
7	Aserțiuni. Depanarea programelor	expunere, dezbateri, studii de caz, rezolvare de probleme	-
8	Recapitulare	expunere, dezbateri, studii de caz, rezolvare de probleme	-
9-10	Măsurarea calității. Metrice pentru calitatea software-ului. Eliminarea defectelor	expunere, dezbateri, studii de caz, rezolvare de probleme	-
11	Modele de fiabilitate pentru software	expunere, dezbateri, studii de caz, rezolvare de probleme	-
12	Metrici de proces pentru testare	expunere, dezbateri, studii de caz, rezolvare de probleme	-
13	Metrici de complexitate	expunere, dezbateri, studii de caz, rezolvare de probleme	-
14	Recapitulare	expunere, dezbateri, studii de caz, rezolvare de probleme	-

Bibliografie**Referințe principale:**

R. D. Craig, S. P. Jaskiel, *Systematic Software Testing*, SQE Publishing, 2007.

S. H. Kahn, *Metrics and Models in Software Quality Engineering*, Second Edition, Addison-Wesley, 2003.

Robert V. Binder, *Testing Object-Oriented Systems: Models, Patterns, and Tools*, Addison-Wesley, 2000.

Referințe suplimentare:

G. J. Myers, *The Art of Software Testing*, Second Edition, Wiley, 2004.

8.2	Seminar / Laborator	Metode de predare	Observații (ore și referințe bibliografice)
1	Testarea programelor; defecte posibile	dezbateri, studii de caz, rezolvare de probleme	-
2	Clase de echivalență	dezbateri, studii de caz, rezolvare de probleme	-
3-5	Unit testing; utilizarea programului NUnit și a bibliotecii NMock	dezbateri, studii de caz, rezolvare de probleme	-
6	Load testing. Stress testing	dezbateri, studii de caz, rezolvare de probleme	-
7	Utilizarea aserțiunilor în testare	dezbateri, studii de caz, rezolvare de probleme	-
8	Recapitulare	dezbateri, studii de caz, rezolvare de probleme	-
9	Lucru la proiect - discutarea temei	rezolvare de probleme	-
10-11	Lucru la proiect - dezvoltarea aplicației	rezolvare de probleme	-
12	Lucru la proiect - proiectarea unit testing	rezolvare de probleme	-
13	Lucru la proiect - utilizarea aserțiunilor	rezolvare de probleme	-



14	Lucru la proiect - elaborarea documentației	rezolvare de probleme	-
----	---------------------------------------------	-----------------------	---

Bibliografie

G. J. Myers, *The Art of Software Testing*, Second Edition, Wiley, 2004.

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

Dezvoltarea proiectelor software de dimensiuni mari este inerent supusă apariției erorilor. Din acest motiv, cunoașterea tehnicilor și metodologiilor de testare este obligatorie pentru managerii de proiect. Dincolo de descoperirea erorilor, este necesară evaluarea eficienței scrierii programelor, precum și stabilirea de măsuri pentru îmbunătățirea acestora, ceea ce implică analiza riscului și aplicarea modelelor statistice asupra evoluției proiectelor software.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere în nota finală (%)
10.4 Curs	înțelegerea calității software la nivel conceptual	test scris	50%
10.5 Seminar/ Laborator	capacitatea de gestionarea dezvoltarea proceselor software de mari dimensiuni	proiect	50%
10.6 Standard minim de performanță			
- înțelegerea fazelor unui proiect software și a etapelor de testare corespunzătoare			
- capacitatea de a realiza un plan de testare pentru un sistem software simplu			
- capacitatea de a utiliza tehnici și unelte de testare (unit testing, aserțiuni, inspecția codului)			

Data completării

Titular de curs

Titular de seminar

Data avizării în departament

Director de departament