



FIȘA DISCIPLINEI COURSE DESCRIPTION

1. Date despre program

Program Information

1.1. Instituția de învățământ superior <i>University</i>	Universitatea “Alexandru Ioan Cuza” din Iași <i>“Alexandru Ioan Cuza” University of Iași</i>
1.2. Facultatea <i>Faculty</i>	Facultatea de Informatică <i>Faculty of Computer Science</i>
1.3. Departamentul <i>Department</i>	Informatică <i>Department of Computer Science</i>
1.4. Domeniul de studii <i>Study Domain</i>	Informatică <i>Computer Science</i>
1.5. Ciclul de studii <i>Study Cycle</i>	Master <i>Undergraduate studies</i>
1.6. Programul de studii / Calificarea <i>Study Program / Qualification</i>	Informatică / Studii universitare de masterat în Ingineria sistemelor software <i>Computer Science / Master graduate in Software Engineering</i>

2. Date despre disciplină

Course Information

2.1. Denumirea disciplinei <i>Course Name</i>	Calitatea sistemelor software <i>Software Quality</i>						
2.2. Titularul activităților de curs <i>Course Teacher</i>	Lect. dr. Vlad Rădulescu						
2.3. Titularul activităților de seminar <i>Seminary Teacher</i>	Lect. dr. Vlad Rădulescu						
2.4. An de studiu <i>Year of study</i>	1	2.5. Semestru <i>Semester</i>	2	2.6. Tip de evaluare <i>Evaluation</i>	E	2.7. Regimul disciplinei <i>Course status</i>	OP

* OB – Obligatoriu / OP – Opțional

* OB – Compulsory / OP – Optional

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

Total estimated hours (hours per semester and didactic activities)

3.1. Număr de ore pe săptămână <i>Hours per week</i>	4	din care: 3.2. curs <i>of which: course</i>	2	3.3. seminar/laborator <i>seminary/laboratory</i>	2
3.4. Total ore din planul de învățământ <i>Hours in curriculum</i>	56	din care: 3.5. curs <i>of which: course</i>	28	3.6. seminar/laborator <i>seminary/laboratory</i>	28
Distribuția fondului de timp <i>Time Distribution</i>					Ore <i>Hours</i>
Studiu după manual, suport de curs, bibliografie și altele <i>Manual study, Course support, Bibliography, and others</i>					35
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren <i>Supplementary Documentation in library, on electronic forums, and in the field</i>					35
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri <i>Seminaries/laboratories preparation, homeworks, reports, portfolios, and essays</i>					70
Tutoriat <i>Tutoring</i>					5
Examinări <i>Evaluation</i>					4
Alte activități <i>Other activities (consultations per student)</i>					5
3.7. Total ore studiu individual <i>Total hours individual study</i>					140



3.8. Total ore pe semestru <i>Total hours per semester</i>	210
3.9. Număr de credite <i>Credits</i>	7

4. Precondiții (dacă este cazul)
Preconditions (if necessary)

4.1. De curriculum <i>Of Curriculum</i>	-
4.2. De competențe <i>Of Skills</i>	-

5. Condiții (dacă este cazul)
Conditions (if necessary)

5.1. De desfășurare a cursului <i>For Course Operation</i>	-
5.2. De desfășurare a seminarului/ laboratorului <i>For Seminary/Laboratory Operation</i>	Prezența este obligatorie la laborator. <i>Attending the seminary/laboratory classes is mandatory.</i>

6. Competențe specifice acumulate
Specific Skills Acquired

Competențe profesionale <i>Professional Skills</i>	La finalizarea cu succes a acestei discipline, studenții vor fi capabili: C1. Să identifice metodologiile adecvate de dezvoltare a sistemelor software. C2. Să identifice modelele și metodele adecvate pentru rezolvarea unor probleme reale. C3. Să proiecteze și să implementeze proiecte informatice dedicate. <i>Upon successful completion of this discipline, students will be able to:</i> C1. Identify the proper methodologies for developing software systems. C2. Identify the proper models and methods for solving real-life problems. C3. Design and implement dedicated informatic projects.
Competențe transversale <i>Transversal Skills</i>	La finalizarea cu succes a acestei discipline, studenții vor fi capabili: C1. Să combine cunoștințele dobândite din diferite surse, să realizeze analogii între problemele și tehnicile de rezolvare din diferite domenii. C2. Să abordeze probleme interdisciplinare. C3. Să utilizeze metode eficiente de învățare, informare, cercetare. <i>Upon successful completion of this discipline, students will be able to:</i> C1. Combine the knowledge acquired from various sources and to make analogies between the problems and solving techniques from various fields. C2. Approach interdisciplinary problems. C3. Use efficient techniques for learning, acquiring information, and research.

7. Obiectivele disciplinei (din grila competențelor specifice acumulate)
Course Objectives (from the grid of specific skills acquired)

7.1. Obiectivul general <i>General Objective</i>	- Înțelegerea principalelor elemente care definesc calitatea sistemelor software. - Familiarizarea cu metodele de testare și analiză a produselor software. <i>- Understanding the main elements that define the quality of software systems.</i> <i>- Getting acquainted with the methods used in program testing and analysis.</i>
--	---



7.2. Obiectivele specifice <i>Specific Objectives</i>	<p>O1. Descrierea principalelor noțiuni legate de testarea software, analiza riscului, planificarea testării, măsurarea calității proiectelor software.</p> <p>O2. Utilizarea de unelte de testare a software-ului.</p> <p>O3. Analiza proiectelor software din punct de vedere al riscului de apariție a defectelor.</p> <p>O4. Planificarea testării sistemelor software.</p> <p>O5. Stabilirea măsurilor care se impun pentru îmbunătățirea procesului de dezvoltare a unui proiect software.</p> <p><i>O1. Describing the main concepts related to software testing, risk analysis, test planning, software quality measurement.</i></p> <p><i>O2. Using software testing tools.</i></p> <p><i>O3. Analyzing software projects and the risk of defect arrival.</i></p> <p><i>O4. Planning the testing of software systems.</i></p> <p><i>O5. Deciding the actions to be taken for improving the development process of a software project.</i></p>
---	---

8. Conținut

General Description

8.1.	Curs <i>Course</i>	Metode de predare <i>Teaching methods</i>	Observații (ore și referințe bibliografice) Observations (hours and bibliographic references)
1	Introducere. <i>Introduction.</i>	expunere, dezbateri, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
2	Testarea programelor. <i>Program testing.</i>	expunere, dezbateri, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
3	Defecte în sistemele software. Inspecția codului. <i>Defects in software systems. Code inspection.</i>	expunere, dezbateri, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
4	Analiza riscului. Planuri de test. <i>Risk analysis. Test planning.</i>	expunere, dezbateri, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
5	Nivele de testare: unit testing, integration testing, system testing, acceptance testing. <i>Testing levels: unit testing, integration testing, system testing, acceptance testing.</i>	expunere, dezbateri, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
6	Testare extremă. Testare regresivă. <i>Extreme testing. Regressive testing.</i>	expunere, dezbateri, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
7	Aserțiuni. Depanarea programelor. <i>Assertions. Debugging.</i>	expunere, dezbateri, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
8	Recapitulare. <i>Recapitulation.</i>	expunere, dezbateri, studii de caz, rezolvare de probleme <i>exposition, debate, case</i>	-



		<i>studies, problem solving</i>	
9	Măsurarea calității. Metrici pentru calitatea software. Eliminarea defectelor. <i>Measuring the software quality. Metrics for software quality. Defect removal.</i>	expunere, dezbatere, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
10	Măsurarea calității. Metrici pentru calitatea software. Eliminarea defectelor. <i>Measuring the software quality. Metrics for software quality. Defect removal.</i>	expunere, dezbatere, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
11	Modele de fiabilitate pentru software. <i>Software reliability models.</i>	expunere, dezbatere, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
12	Metrici de proces pentru testare. <i>Process metrics for testing.</i>	expunere, dezbatere, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
13	Metrici de complexitate. <i>Complexity metrics.</i>	expunere, dezbatere, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
14	Recapitulare. <i>Recapitulation.</i>	expunere, dezbatere, studii de caz, rezolvare de probleme <i>exposition, debate, case studies, problem solving</i>	-
Bibliografie R. D. Craig, S. P. Jaskiel, <i>Systematic Software Testing</i> , SQE Publishing, 2007. S. H. Kahn, <i>Metrics and Models in Software Quality Engineering</i> , Second Edition, Addison-Wesley, 2003. Robert V. Binder, <i>Testing Object-Oriented Systems: Models, Patterns, and Tools</i> , Addison-Wesley, 2000. G. J. Myers, <i>The Art of Software Testing</i> , Second Edition, Wiley, 2004. Bibliography R. D. Craig, S. P. Jaskiel, <i>Systematic Software Testing</i> , SQE Publishing, 2007. S. H. Kahn, <i>Metrics and Models in Software Quality Engineering</i> , Second Edition, Addison-Wesley, 2003. Robert V. Binder, <i>Testing Object-Oriented Systems: Models, Patterns, and Tools</i> , Addison-Wesley, 2000. G. J. Myers, <i>The Art of Software Testing</i> , Second Edition, Wiley, 2004.			
8.2.	Seminar / Laborator <i>Seminary / Laboratory</i>	Metode de predare <i>Teaching methods</i>	Observații (ore și referințe bibliografice) Observations (hours and bibliographic references)
1	Testarea programelor. Defecte posibile. <i>Program testing. Possible defects.</i>	dezbatere, studii de caz, rezolvare de probleme <i>debate, case studies, problem solving</i>	-
2	Clase de echivalență. <i>Equivalence classes.</i>	dezbatere, studii de caz, rezolvare de probleme <i>debate, case studies, problem solving</i>	-
3	Unit testing; utilizarea uneltelor de unit testing și de mocking. <i>Unit testing; using unit testing and mocking tools.</i>	dezbatere, studii de caz, rezolvare de probleme <i>debate, case studies, problem solving</i>	-
4	Unit testing; utilizarea uneltelor de unit testing și de mocking. <i>Unit testing; using unit testing and</i>	dezbatere, studii de caz, rezolvare de probleme <i>debate, case studies, problem</i>	-



	<i>mocking tools.</i>	<i>solving</i>	
5	Unit testing; utilizarea uneltelor de unit testing și de mocking. <i>Unit testing; using unit testing and mocking tools.</i>	dezbateri, studii de caz, rezolvare de probleme <i>debate, case studies, problem solving</i>	-
6	Load testing. Stress testing. <i>Load testing. Stress testing.</i>	dezbateri, studii de caz, rezolvare de probleme <i>debate, case studies, problem solving</i>	-
7	Utilizarea aserțiunilor în testare. <i>Using assertions in the testing process.</i>	dezbateri, studii de caz, rezolvare de probleme <i>debate, case studies, problem solving</i>	-
8	Recapitulare. <i>Recapitulation.</i>	dezbateri, studii de caz, rezolvare de probleme <i>debate, case studies, problem solving</i>	-
9	Lucru la proiect - discutarea temei. <i>Project work - specifications.</i>	rezolvare de probleme <i>problem solving</i>	-
10	Lucru la proiect - dezvoltarea aplicației. <i>Project work - application development.</i>	rezolvare de probleme <i>problem solving</i>	-
11	Lucru la proiect - dezvoltarea aplicației. <i>Project work - application development.</i>	rezolvare de probleme <i>problem solving</i>	-
12	Lucru la proiect - unit testing. <i>Project work - unit testing.</i>	rezolvare de probleme <i>problem solving</i>	-
13	Lucru la proiect - utilizarea aserțiunilor. <i>Project work - assertions.</i>	rezolvare de probleme <i>problem solving</i>	-
14	Lucru la proiect - elaborarea documentației. <i>Project work - creation of the documentation.</i>	rezolvare de probleme <i>problem solving</i>	-
Bibliografie G. J. Myers, <i>The Art of Software Testing</i> , Second Edition, Wiley, 2004. Bibliography G. J. Myers, <i>The Art of Software Testing</i> , Second Edition, Wiley, 2004.			

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

Course content synchronization with the expectations of the community representatives, professional associations and employers from the program domain

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.

The development of large software projects is an inherently error-prone activity. That is why knowing the testing techniques and methodologies is mandatory for the project managers. Beyond error tracking, the efficiency of program writing must be assessed and appropriate measures must be taken to improve the process; to achieve that, risk analysis and statistical models must be used in order to predict/improve the evolution of software projects.

10. Evaluare

**Evaluation**

Tip activitate <i>Activity Type</i>	10.1.Criterii de evaluare <i>Evaluation criteria</i>	10.2.Metode de evaluare <i>Evaluation methods</i>	10.3.Pondere în nota finală (%) <i>Weight per evaluation form (%)</i>
10.4.Curs <i>Course</i>	Înțelegerea calității software la nivel conceptual. <i>Understanding the concepts related to software quality.</i>	1 test scris <i>1 written test</i>	50%
10.5.Seminar/ Laborator <i>Seminary/ Laboratory</i>	Capacitatea de gestionarea dezvoltării proceselor software de mari dimensiuni. <i>The ability to handle the development of large software projects.</i>	1 proiect <i>1 project</i>	50%
10.6.Standard minim de performanță <i>Minimal performance standards</i>			
<ul style="list-style-type: none">➤ 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).➤ <i>Understanding the phases of a software project and the corresponding testing phases.</i>➤ <i>The ability to develop a test plan for a simple software system.</i>➤ <i>The ability to use testing techniques and tools (unit testing, assertions, code inspection).</i>			

Data completării
Date

Titular de curs și seminar
*Course and Seminary/Laboratory
Teacher*

30.09.2023

Data avizării în departament
Department Date of Approval

Director de departament
Director of the Department