

**FIȘA DISCIPLINEI**  
**Dezvoltarea sistemelor de programe**  
*(Software Engineering)*  
 UP.01.DA.01.O.34.03

**1. Date despre program**

1.1	Instituția de învățământ superior	Universitatea din Pitești
1.2	Facultatea	Științe, Educație Fizică și Informatica
1.3	Departamentul	Matematica-Informatica
1.4	Domeniul de studii	Informatica
1.5	Ciclul de studii	Master
1.6	Programul de studii / Calificarea	<b>Tehnici avansate pentru procesarea informației (în limba engleză)</b> ( <i>Advanced Techniques for Information Processing</i> ) / Informatică

**2. Date despre disciplină**

2.1	Denumirea disciplinei	<b>Dezvoltarea sistemelor de programe (Software Engineering)</b>									
2.2	Titularul activităților de curs	Prof. univ. dr. Tudor Bălănescu									
2.3	Titularul activităților de laborator	Prof. univ. dr. Tudor Bălănescu									
2.4	Anul de studii	1	2.5	Semestrul	1	2.6	Tipul de evaluare	Examen	2.7	Regimul disciplinei	O

**3. Timpul total estimat**

3.1	Număr de ore pe săptămână	4	3.2	din care curs	2	3.3	laborator	2
3.4	Total ore din planul de inv.	56	3.5	din care curs	28	3.6	laborator	28
Distribuția fondului de timp								ore
Studiul după manual, suport de curs, bibliografie și notițe								54
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren								40
Pregătire seminarii/laboratoare, teme, referate, portofolii, eseuri								40
Tutoriat								6
Examinări								4
Alte activități								-
3.7	Total ore studiu individual			144				
3.8	<b>Total ore pe semestru</b>			<b>200</b>				
3.9	<b>Număr de credite</b>			<b>8</b>				

**4. Precondiții (acolo unde este cazul)**

4.1	De curriculum	-
4.2	De competențe	-experiență de programare

**5. Condiții (acolo unde este cazul)**

5.1	De desfășurare a cursului	Sală dotată cu videoprojector
5.2	De desfășurare a laboratorului	Laborator de informatică

**6. Competențe specifice acumulate**

Competențe profesionale	-abilitatea de a descrie funcțional sisteme software și a elabora documentație de tip "cerințe utilizator"; -aptitudine de a elabora și de a specifica formal proiecte de sisteme software; utilizând limbaje și formalisme specializate (UML, Z, rețele Petri, statecharts, mașini cu număr finit de stări, automate cu cronometru) -îndemânare în utilizarea procedurilor de verificare și validare a sistemelor (testare, generare automata a testelor) -experiență în utilizarea sistemelor specializate pentru verificarea modelelor (model checking)
Competențe transversale	Înșușirea unor elemente de etică a ingineriei programării; Coordonarea și conducerea eficientă a activităților organizate în colectivele de elaborare a sistemelor software; Manifestarea unei atitudini responsabile față de domeniul științific și didactic, valorificarea potențialului propriu pe plan profesional, respectarea regulilor de muncă riguroasă și eficiență pentru executarea unor sarcini profesionale complexe;

**7. Obiectivele disciplinei**

7.1	Obiectivul general al disciplinei	Disciplina are ca obiectiv general însușirea de către studenți a cunoștințelor și metodelor privind ingineria software și utilizarea unor aplicații de modelare UML și aplicarea acestor instrumente în rezolvarea unor probleme specifice.
7.2	Obiectivele specifice	Obiective cognitive: Învățarea și însușirea conceptelor de bază la disciplina „software engineering”. Obiective procedurale: elaborarea unui plan de dezvoltare a unui produs software, trecând prin toate fazele de dezvoltare de la elaborarea cerințelor, modelarea sistemului,

	implementarea si testarea, utilizarea unor pachete si produse software care sustin activitatile de mai sus, in special modelarea cu limbajul UML, aplicarea pasilor de mai sus in cadrul unui proiect concret, elaborat de student sub indrumarea titularului de laborator. Obiective atitudinale: Rigurozitate în proiectarea și implementarea algoritmilor.
--	--

## 8. Conținuturi

8.1. Curs		Metode de predare	Observații Resurse folosite
1	Introduction – 2 ore <ul style="list-style-type: none"> <li>Professional software development</li> <li>Software engineering ethics</li> <li>Case studies</li> </ul>	Expunerea cu material suport. Explicația. Descrierea și exemplificarea	Calculator, Videoprojector
2	Software Processes – 2 ore <ul style="list-style-type: none"> <li>Software process models</li> <li>Process activities</li> <li>Coping with change</li> <li>The Rational Unified Process</li> </ul>		
3	Agile software development – 2 ore <ul style="list-style-type: none"> <li>Agile methods</li> <li>Plan-driven and agile development</li> <li>Extreme programming</li> <li>Agile project management</li> <li>Scaling agile methods</li> </ul>		
4	Requirements engineering – 2 ore <ul style="list-style-type: none"> <li>Functional and non-functional requirements</li> <li>The software requirements document</li> <li>Requirements specification</li> <li>Requirements engineering processes</li> <li>Requirements elicitation and analysis</li> <li>Requirements validation</li> <li>Requirements management</li> </ul>		
5	System modeling – 6 ore <ul style="list-style-type: none"> <li>Context models</li> <li>Interaction models</li> <li>Structural models</li> <li>Behavioral models</li> <li>Model-driven engineering</li> </ul>		
6	Architectural design – 4 ore <ul style="list-style-type: none"> <li>Architectural design decisions</li> <li>Architectural views</li> <li>Architectural patterns</li> <li>Application architectures</li> </ul>		
7	Design and Implementation – 4 ore <ul style="list-style-type: none"> <li>Object-oriented design using the UML</li> <li>Design patterns</li> <li>Implementation issues</li> <li>Open source development</li> </ul>		
8	Software testing – 4 ore <ul style="list-style-type: none"> <li>Development testing</li> <li>Test-driven development</li> <li>Release testing</li> <li>User testing</li> <li>Unit, Integration, System testing</li> </ul>		
9	Other topics – 2 ore <ul style="list-style-type: none"> <li>Software Evolution</li> <li>Software Reuse</li> <li>Distributed Software Engineering</li> <li>Service-oriented Architecture</li> </ul>		
<p>Bibliografie:</p> <ul style="list-style-type: none"> <li>Ian Sommerville. Software Engineering – 9th edition (2010) - Addison-Wesley</li> <li>G. Booch, J. Rumbaugh, I. Jacobson. The Unified Language User Guide, Addison-Wesley, 1999.</li> <li>M. Fowler, K. Scott. UML Distilled: Applying the Standard Object Modeling Language, Addison-Wesley, 1997.</li> <li>F. Ipate. Modelare orientata pe obiecte cu UML, Editura Universitatii Pitesti, 2001.</li> <li>M. Roper. Software Testing, McGraw-Hill, 1994.</li> <li>M. Holcombe, F. Ipate. Correct Systems: Building a Business Process Solution, Springer Verlag, 1998.</li> <li>Tudor Bălănescu, Horia Georgescu, Marian Gheorghe, Peter O'Donogue: "HOOD and Regular Expressions", Analele Universității București, Seria Matematică-Informatică, Special Issue, Proceedings of the Annual</li> </ul>			

Meeting of the Faculty of Mathematics, 28-39 Nov. 1996, p. 45-60, 1997.

8.2. Aplicații – Laborator		Metode de predare	Observații Resurse folosite
1	<ul style="list-style-type: none"> <li>Proiect software care sa reflecte fidel notiunile si etapele de dezvoltare software predate la curs</li> <li>Exemple pentru notiunile predate la curs</li> <li>Prezentarea de tool-uri pentru modelare UML (d.ex. Magic Draw, ArgoUML)</li> </ul> (4 ore)	Problematizarea  Exercițiul, Studiul de caz, Dezbateră	Rețea de calculatoare, Internet
2	<ul style="list-style-type: none"> <li>Specificare a cerințelor de utilizare și elaborare a unui proiect, descris în UML</li> </ul> (4 ore)		
3	<ul style="list-style-type: none"> <li>Diagrame de utilizare, diagrame de stare, diagrame de secvență</li> </ul> (4 ore)		
4	<ul style="list-style-type: none"> <li>Exemplu de specificare formală a unui sistem real time</li> </ul> (4 ore)		
5	<ul style="list-style-type: none"> <li>Specificare formală a proprietăților și verificare a modelelor (model checking)</li> </ul> (4 ore)		
6	<ul style="list-style-type: none"> <li>Exerciții de testare funcțională</li> </ul> (4 ore)		
7	<ul style="list-style-type: none"> <li>Exerciții de testare structurală</li> </ul> (4 ore)		
Bibliografie <ul style="list-style-type: none"> <li>Ian Sommerville. Software Engineering – 9th edition (2010) - Addison-Wesley</li> <li>UML tool documentations</li> <li>F. Ipate. Modelare orientata pe obiecte cu UML, Editura Universitatii Pitesti, 2001.</li> </ul>			

**9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori din domeniul aferent programului**

Competențele dobândite la disciplină permit absolvenților să utilizeze eficient pachete software pentru indeplinirea sarcinilor aferente practicii sau cercetării în domeniul ingineriei software.

**10. Evaluare**

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Evaluare finală	Examen de verificare a cunoștințelor de inginerie software	50%
10.5 Seminar/ Laborator	Proiect software care să reflecte noțiunile și etapele de dezvoltare software predate la curs	Demonstrație practică a sistemului dezvoltat	50%
10.6 Standard minim de performanță	Note de minim 5 la curs, si activitatea de laborator; nota finală minim 5.		

Data completării  
20.09.2020

Titular de curs  
Prof. univ. dr. Tudor BĂLĂNESCU

Titular de laborator  
Prof. univ. dr. Tudor BĂLĂNESCU

Data avizării în Departament

Director Departament (*prestator*)  
Conf.univ.dr. Doru CONSTANTIN

Director Departament (*beneficiar*)  
Conf.univ.dr. Doru CONSTANTIN