FIŞA DISCIPLINEI[1]

1. Date despre program

1.1 Instituţia de învăţământ superior	Universitatea „Politehnica” din Timişoara
1.2 Facultatea[2] / Departamentul[3]	Automatică şi Calculatoare / Calculatoare
1.3 Catedra	-
1.4 Domeniul de studii	Calculatoare şi Tehnologia informaţiei
1.5 Ciclul de studii	Licenţă
1.6 Programul de studii / Calificarea	Calculatoare / inginer

2. Date despre disciplină

2.1 Denumirea disciplinei			Fundamente de Inginerie Software				
2.2 Titularul activităţilor de curs			Prof.dr.ing. habil. Radu Marinescu				
2.3 Titularul activităţilor de seminar			Șl.dr.ing. Petru Florin Mihancea, Șl.dr.ing. Codruța Istin, Drd.ing. Cosmin Marșavina				
2.4 Anul de studiu	2	2.5 Semestrul	4	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatorie

3. Timpul total estimat (ore pe semestru al activităţilor didactice)

3.1 Număr de ore pe săptămână	5	din care:3.2 curs	2	3.3 seminar/laborator	3
3.4 Total ore din planul de învăţământ	121	din care:3.5 curs	28	3.6 seminar/laborator	42
Distribuţia fondului de timp					Ore
Studiul după manual, suport de curs, bibliografie şi notiţe					19
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren					12
Pregătire seminarii/laboratoare, teme, referate, portofolii şi eseuri					20
Tutoriat					5
Examinări					4
Alte activităţi					

3.7 Total ore studiu individual	51
3.8 Total ore pe semestru	130
3.9 Numărul de credite	5

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	• Programarea orientata pe obiecte
4.2 de competenţe	• Programare orientata pe obiecte

5. Condiţii (acolo unde este cazul)

5.1 de desfăşurare a cursului	• Sală mare, Materiale suport: laptop, proiector, tablă.
5.2 de desfăşurare a seminarului/laboratorului	• Laborator cu 15-20 calculatoare – tablă

[1] Formularul corespunde Fişei Disciplinei promovată prin OMECTS 5703/18.12.2011 (Anexa3);

[2] Se înscrie numele facultăţii care gestionează programul de studiu căruia îi aparţine disciplina;

[3] Se înscrie numele departamentului căruia i-a fost încredinţată susţinerea disciplinei şi de care aparţine titularul cursului;

6. Competenţe specifice acumulate

Competenţe profesionale[4]	• Operarea cu fundamente ştiinţifice, inginereşti şi ale informaticii • Proiectarea componentelor hardware, software şi de comunicaţii • Soluţionarea problemelor folosind instrumentele ştiinţei şi ingineriei calculatoarelor • Îmbunătăţirea performanţelor sistemelor hardware, software şi de comunicaţii • Proiectarea, gestionarea ciclului de viaţă, integrarea şi integritatea sistemelor hardware, software şi de comunicaţii • Proiectarea sistemelor inteligente
Competente transversale	• Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura rezolvarea problemei • Indentificarea, descrierea şi derularea proceselor din managementul proiectelor, cu preluarea diferitelor roluri în echipă şi descrierea clară şi concisă, verbal şi în scris, în limba română şi într-o limbă de circulaţie internaţională, a rezultatelor din domeniul de activitate • Demonstarea spiritului de iniţiativă şi acţiune pentru actualizarea cunoştinţelor profesionale, economice şi de cultură organizaţională

7. Obiectivele disciplinei (reieşind din grila competenţelor specifice acumulate)

7.1 Obiectivul general al disciplinei	Cursul prezinta principalele concepte, metode si tehnici ale ingineriei software, cu accent pe orientarea obiectuală. La sfârșitul semestrului studenții trebuie să poată aprecia importanța dezvoltării de produse software aplicând metode inginerești, pentru a ajunge la produse de calitate, economice și livrate la timp.
7.2 Obiectivele specifice	• Familiarizarea cu diferite procese de dezvoltare a software-ului, cu înțelegerea diferitelor avantaje și dezavantaje ale fiecăruia • Cunoașterea unor tehnici fundamentale de captare a cerințelor • Acomodarea cu principalele tipuri de diagrame prin care se modelează un sistem software • Înțelegerea scopului verificării și validării sistemelor software, precum și a principalelor tehnici și instrumente de testare

8. Conţinuturi

8.1 Curs	Număr de ore	Metode de predare
1.Introducere 1.1 Scopul și obiectivele ingineriei software 1.2 Specificitățile sistemelor software 1.3 Etapele ciclului de viata software 1.4 Miturile ingineriei software	4	Prelegere susţinută de prezentări Keynote, conversaţii, explicaţii, exemplificări
2. Procese de dezvoltare a sistemelor software 2.1 Procesul Waterfall 2.2 Procesele iterative și incrementale 2.3 Progese agile 2.4 Procesul de dezvoltare RUP 2.5 Prototipizarea	6	
3. Ingineria cerintelor 3.1 Cerinte funcționale și nefuncționale 3.2 Procesul de inginerie a cerintelor 3.3 Captarea cerințelor folosing *Use Case.*	6	

[4] Aspectul competenţelor profesionale va fi tratat cf. Metodologiei OMECTS 5703/18.12.2011. Se vor prelua competenţele care sunt precizate în Registrul Naţional al Calificărilor din Învăţământul Superior RNCIS (http://www.rncis.ro/portal/page?_pageid=117,70218&_dad=portal&_schema=PORTAL) pentru domeniul de studiu de la pct. 1.4 şi programul de studii de la pct. 1.6 din această fişă.

4. Modelarea și Proiectarea Sistemelor Software 4.1. Modelarea sistemelor software 4.2 Diagramele de clase 4.3 Diagramele de secventa 4.4 Proiectarea arhitecturala 4.5 Stiluri arhitecturale 4.6 Proiectarea interfeței cu utilizatorul.	6	
5. Verificarea și Validarea 5.1. Inspecții software 5.2 Principii de testare a produselor software 5.3 Testarea unitara și testarea de integrare 5.4 Tehnici de testare *blackbox* și *whitebox* 5.5 Validarea sistemelor software	6	

Bibliografie

1. **Ian Sommerville**, *Software Engineering 8th Edition;* Addison-Wesley, 2006
2. **Roger S. Pressman**, *Software Engineering: A Practitioner's Approach, 6th Edition;* Editura McGraw-Hill, 2004
3. **Steve McConnell**, *Code Complete: A Practical Handbook of Software Construction, 2nd Edition;* Editura Microsoft Press, 2004

8.2 Seminar/laborator	Număr de ore	Metode de predare
Laboratorul va consta din urmatoarele teme: • Construirea de diagrame UML de clase respectiv diagrame de secvență și exersarea mapării între cod și diagrame • Sisteme de versionare SVN: principii, comenzi de bază • Sisteme de Build Ant / Makefile: principii, build file, exemplu. Temă: de scris un fisier simplu de build, cu un target de build și unul de deploy proiect simplu dat, cu dependențe către biblioteci. • Lucrul cu un mediu integrat de dezvoltare (IDE): integrare CVS, Ant, code conventions, code templates, refactorings, etc. (2 ore) • Automatizarea testarii cu jUnit/nUnit. Tema: sa scrie si sa ruleze un nr de teste pt o clasa data accent pe tool si pe automatizare • Ingineria cerintelor: identificarea de use cases, actori, descriere use-case (story cards); diagrame UML de use-case. Tema: identificare de actori și elaborarea unei use case diagram dintr-o descriere text dată. Descrierea unui use cases sub forma de story cards. Proiectul va exersa toate conceptele studiate, conducându-l pe student prin toate fazele procesului de dezvoltare inclusiv negocierea de use-case-urilor de implementat intr-o iteratie	42	Expunere temă, discuţii, întrebări.

Bibliografie

1. **Martin Fowler** - *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd Edition, Addison-Wesley, 2003
2. **Andy Hunt**, **Dave Thomas** - *Pragmatic Unit Testing in Java with JUnit*; Pragmatic Programmers, 2003
3. **Mike Clark** - *Pragmatic Project Automation: How to Build, Deploy, and Monitor Java Apps*; Pragmatic Programmers, 2004
4. **Mike Mason** - *Pragmatic Version Control: Using Subversion*; Pragmatic Programmers, 2006

9. Corelarea conţinutului disciplinei cu cerinţele specialiştilor din domeniu şi cu aşteptările angajatorilor reprezentativi

- Cunoştinţele elementelor fundamentale ale ingineriei software sunt esențiale pentru dobândirea abilității de a funcționa în cadrul oricărei companii specializate în dezvoltarea de software.
- Majoritatea angajatorilor reprezentativi din domeniul aferent programului solicită cunoştinţe de modelare, testare, și proiectare a sistemelor software.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Expunerea diverselor subiecte prezentate la curs Aplicabilitatea practică a noțiunilor însușite	Examinare scrisă	67 %
10.5 Seminar /laborator		Teste scrise ce acoperă prin întrebări principalele teme abordate la laborator.	33%
10.6 Standard minim de performanţă (volumul de cunoştinţe minim necesar pentru promovarea disciplinei şi modul în care se verifică stăpânirea lui)			
• Capacitatea de a transcrie în cod o diagramă de clasă și de secvență • Capacitatea de a construi diagrame de clase și de secvență dintr-un fragment de cod • Capacitatea de scrie secvențe de date de test pentru o testare de tip blackbox respectiv whitebox • Capacitatea de a construi o diagramă de tip use-case dintr-o specificare de cerințe			

11. Compatibilitate internaţională

1. ***Universitatea din Calgary*** *(Canada): http://bit.ly/1eB8SVG*
2. ***ETH Zurich*** *(Elveția): http://bit.ly/1b5RF6g*
3. ***University of Illinois at Urbana-Champaign*** *(USA): http://bit.ly/1ah8s1B*

Data completării	Semnătura titularului de curs	Semnătura titularilor de seminar
16.09.2014	Prof.dr.ing.habil Radu Marinescu	Șl.dr.ing. Petru F. Mihancea, Șl.dr.ing. Codruța Istin,
	.....	.....
		Drd.ing. Cosmin Marșavina
		.....

Data avizării în departament	Semnătura directorului de departament
	Prof.dr.ing. Vladimir Ioan Crețu
	.....