

FIȘA DISCIPLINEI¹

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea „Politehnica” din Timișoara
1.2 Facultatea ² / Departamentul ³	Automatică și Calculatoare / Automatică și Informatică Aplicată
1.3 Catedra	-
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licență - Învățământ la distanță
1.6 Programul de studii / Calificarea	Informatică / Informatician

2. Date despre disciplină

2.1 Denumirea disciplinei	Fundamente de inginerie software						
2.2 Titularul activităților de curs	Prof. dr. ing. Vasile Stoicu-Tivadar						
2.3 Titularul activităților de seminar	Prof. dr. ing. Vasile Stoicu-Tivadar						
2.4 Anul de studiu	2	2.5 Semestrul	2	2.6 Tipul de evaluare	D	2.7 Regimul disciplinei	Obligatorie

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

3.1 Număr de ore pe săptămână	4	din care:3.2 curs (SI)	2	3.3 seminar/laborator (AA)	2
3.4 Total ore din planul de învățământ	56	din care:3.5 curs (SI)	28	3.6 seminar/laborator (AA)	28
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					24
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					17
Tutoriat					12
Examinări					6
Alte activități					
3.7 Total ore studiu individual	74				
3.8 Total ore pe semestru	130				
3.9 Numărul de credite	5				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Introducere in programarea calculatoarelor. Tehnici de programare. Structuri de date si algoritmi. Programare orientată pe obiecte
4.2 de competențe	<ul style="list-style-type: none"> Nu este cazul

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	<ul style="list-style-type: none"> Sală cu ecran și videoproiector, laptop
5.2 de desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> Sală cu tehnică de calcul (1-2 studenți/calculator), ecran, videoproiector Masă “de consiliu” (masă mare, în jurul căreia pot sta 10 studenți)

6. Competențe specifice acumulate

Competențe profesionale ⁴	Programarea in limbaje de nivel inalt Dezvoltarea si intretinerea aplicatiilor informatice Utilizarea instrumentelor informatice in context interdisciplinar
--------------------------------------	--

¹ Formularul corespunde Fișei Disciplinei promovată prin OMECTS 5703/18.12.2011 (Anexa3);

² Se înscrie numele facultății care gestionează programul de studiu căruia îi aparține disciplina;

³ Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului;

⁴ Aspectul competențelor profesionale va fi tratat cf. Metodologiei OMECTS 5703/18.12.2011. Se vor prelua competențele care sunt precizate în Registrul Național al Calificărilor din Învățământul Superior RNCIS (http://www.rncis.ro/portal/page?_pageid=117_70218&_dad=portal&_schema=PORTAL) pentru domeniul de studiu de la pct. 1.4, programul de studii de la pct. 1.6 din această fișă și materia în cauză

Competențe transversale	
-------------------------	--

7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1 Obiectivul general al disciplinei	Însușirea principalelor concepte care operează în cadrul activității de proiectare software, respectiv însușirea cunoștințelor și deprinderilor practice necesare lucrului într-o echipă de proiectare software.
7.2 Obiectivele specifice	Dobândirea unor cunoștințe pentru abordarea cu succes a proiectelor software complexe cu exemple din domeniul sistemelor dedicate/înglobate. Prin studierea/dezvoltarea unor sisteme industriale din diverse domenii se va urmări întărirea competențelor pentru dezvoltarea și întreținerea aplicațiilor informatice precum și pentru utilizarea instrumentelor informatice în context interdisciplinar.

8. Conținuturi

8.1 Curs (SI – studiu individual)	Număr de ore	Metode de predare
Introducere în ingineria programării (Definiții, exemple, noțiuni de bază: obiective, principii, ciclul de viață al programelor: prototipizarea, ciclul în V, modelul spirală, metoda incrementală, metoda iterativă, dezvoltarea AGILE. Standarde)	5	Cursul va fi prezentat pe suport ppt, va fi interactiv/problematicizat, se prezintă studii de caz, se analizează scenarii etc.
Ingineria cerințelor (Ce sunt cerintele. Cerinte utilizator. Cerinte sistem. Cerinte functionale. Cerinte non-functionale. Cerinte de domeniu)	2	
Proiectarea sistemelor software (Arhitectura unui sistem software. Modele cu zona de date partajata, client-server, three-tier, masina abstracta. Descompunerea in module. Programarea structurată - caracteristici, cuplarea modulelor, principii euristice, dezvoltarea software - specificații, proiectare, codificare, testare, asistență tehnică)	4	Materialele suport de curs precum și materialele privind metodologia de proiectare, tema de proiectare, alte materiale suplimentare, vor fi disponibile în format electronic pe campus virtual
Testarea și validarea software (Testarea programului. Inspectia codului. Fazele testarii. Testarea black box. Teste de performanta. Automatizarea testarii.)	2	
Siguranța în funcționare (Generalități, prevenirea defectelor, redundanța statică și dinamică, restabilirea în toleranța la defectare, tratarea excepțiilor)	2	
Sisteme informatice industriale (Sisteme informatice pentru conducerea proceselor - structuri, funcții, sisteme de operare utilizate, sisteme înglobate, metoda MASOCOT de proiectare a aplicațiilor în timp real)	2	
Managementul proiectelor software. Gestiunea configurațiilor (Management de proiecte. Specificul proiectelor software. Comunicarea. Planificarea, controlul și estimarea efortului de proiectare. Calitatea software și standarde de calitate. Problematika gestiunii configurațiilor - probleme, modalități de rezolvare, instrumente suport)	5	
UML în timp real (Analiza cerințelor, definirea structurii și comportării, proiectarea pe 3 nivele, tipuri de diagrame UML, exemple. Sabloane de proiectare, definiții, exemple)	6	
Bibliografie 1. Stoicu-Tivadar, V., „Fundamente de inginerie software”, format electronic, campus virtual http://elearning.upt.ro , Timișoara, 2012 2. Pescaru, Dan, „Fundamente de inginerie software”, 2010. http://www.cs.upt.ro/~dan/ 3. Thayer, R., Christenes, M.J., Software Engineering. The development process”IEEE Press, 2005 4. Coplan, S., Masuda, D., „Project management for Healthcare Information Technology”, McGrawHill, 2011 5. Pflieger, S.L., “Software Engineering. Theory and Practice”, Prentice Hall, 1998 6. Bărbat, B., Filip, F.Gh., „Informatica Industrială. Ingineria programării în timp real”, Editura tehnică, București, 1997		
8.2 Seminar/laborator (AA – activități asistate)	Număr de ore	Metode de predare
Proiect. Specificații (Fluxul proiectării în cadrul unui colectiv software. Organizarea echipei de proiectare (metoda echipei programatorului-șef). Elaborarea	8	La activitatea pe parcurs, se va parcurge întregul ciclu de viață al

specificațiilor)		programelor, prin activitate în cadrul unei echipe de proiectare software (la nivelul unei semigrupe). Fiecare semigrupă va primi o temă de proiectare și va acționa ca o echipă care lucrează la o companie software, intrând pe cât posibil mai bine în rol. Se va aplica metoda de organizare a „echipei programatorului-șef”. Cadrul didactic îndrumător va juca alternativ rolul propriu, pe cel al clientului, pe cel al directorului companiei, respectiv al șefului compartimentului de calitate.
Laborator. Utilizarea unor instrumente suport (Utilizarea mediului Microsoft Project pentru planificarea activităților. Utilizarea WinCVS pentru managementul configurațiilor. Utilizarea StarUML)	6	
Proiect. Proiectarea și testarea (Proiectarea de detaliu. Caietul de sarcini. Testarea aplicației)	8	
Proiect. Finalizarea proiectului software (Realizarea unui prototip funcțional Prezentarea unui proiect pentru avizare)	6	
<p>Bibliografie</p> <p>1 Stoicu-Tivadar, V., „Fundamente de inginerie software”, format electronic, campus virtual http://elearning.upt.ro, Timișoara, 2012</p> <p>2. Coplan, S., Masuda, D., „Project management for Healthcare Information Technology”, McGrawHill, 2011.</p>		

9. Corelarea conținutului disciplinei cu cerințele specialiștilor din domeniu și cu așteptările angajatorilor reprezentativi

Toți angajatorii reprezentativi din domeniul aferent programului solicită atât cunoștințe de programare în general cât și abilități de lucru în echipă și cunoștințe elementare despre procesul de dezvoltare software, sub diverse forme, respectiv cunoașterea unor unelte și metodologii utile în acest context, în special pentru dezvoltarea sistemelor înglobate.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs (SI - studiu individual)	<p>Evaluare teoretică, cu unele întrebări problematizate (trimiteri la cunoștințe practice).</p> <p>Nota se obține pe baza proporționalității răspunsurilor corecte (toate răspunsurile corecte corespund la nota maximă).</p>	La verificarea cunoștințelor : se răspunde la un test-grilă cu 50-70 întrebări (1 min/întrebare), test cu variante unice de răspuns. Unele întrebări sunt problematizate, adică necesită înțelegerea unor noțiuni de la curs și aplicarea lor corectă în diverse contexte cvasi-reale.	66 %
10.5 Seminar /laborator (AA – activități asistate)	<p>Evaluarea practică, a încadrării în activitatea unei echipe de dezvoltare software.</p> <p>Nota este negociată între membrii echipei pe baza aportului fiecăruia (timp alocat, eficiență, pondere în proiect, dificultate, importanță pentru rezultatul final)</p>	La activitatea pe parcurs : se acordă un punctaj final total pentru toată semigrupa (echipa) în urma susținerii finale și membrii echipei împart punctajul între ei, în funcție de aportul fiecăruia, definitivat prin negocieri în cadrul echipei.	34 %
<p>10.6 Standard minim de performanță (volumul de cunoștințe minim necesar pentru promovarea disciplinei și modul în care se verifică stăpânirea lui). Prin specificul testului grilă de a baleia întreaga materie, condiția de promovare presupune însușirea la nivel satisfăcător a cunoștințelor necesare pentru a lucra în echipe de dezvoltare software și a cunoaște la nivel elementar problematica managementului de proiect software, respectiv cunoașterea la nivel elementar a unor aspecte de proiectare orientată pe obiecte și limbaj UML. Nota de promovare minimă este 5 (respectiv, răspuns corect la jumătate din întrebări).</p> <p>Prin participarea la proiectul de echipă și ulterior prin negocierea notei în cadrul echipei, fiecare membru trebuie să se dovedească util într-un fel (altfel colegii nu îi alocă punctaj de promovare) și deci este atins obiectivul ca studenții să se facă utili la nivel măcar</p>			

satisfăcător, pentru realizarea unui proiect software. Nota minimă de promovare este 5.

11. Compatibilitate internațională

Practic nu se poate concepe pregătire în informatică fără „Ingineria programării” („Software engineering” sau similare). Exemple:

1. Cornell University, <http://www.cs.cornell.edu>
Software Engineering <http://www.cs.cornell.edu/courses/cs501/2005sp/syllabus.html>
2. Florida State University <http://www.cs.fsu.edu>
Software Engineering <http://www.cs.fsu.edu/~gaitrosd/classes/Summer2004/cen4010/syllabus.html>
3. MIT <http://www.mit.edu>
Software Engineering <http://courses.csail.mit.edu/6.170/old-www/2001-Fall/info/syllabus.pdf>

Data
completării

05.02.2015

Semnătura titularului de curs

Prof. dr. ing. Vasile STOICU-
TIVADAR

Semnătura titularilor de seminar

Prof. dr. ing. Vasile STOICU-TIVADAR

Data avizării în departament

05.02.2015

Semnătura directorului de departament

Prof. dr. ing. Ioan SILEA