FIŞA DISCIPLINEI

1. Date despre program

1.1. Instituţia de învăţământ superior	Universitatea „Sapientia" din municipiul Cluj-Napoca
1.2. Facultatea	Facultatea de Ştiinţe Tehnice şi Umaniste din Târgu Mureş
1.3. Domeniul de studii	Calculatoare şi tehnologia informaţiei
1.4. Ciclul de studii	Licenţă
1.5. Programul de studiu	Calculatoare
1.6. Calificarea	Inginer

2. Date despre disciplină

2.0. Departamentul				Departamentul de Inginerie Electrică			
2.1. Denumirea disciplinei				Testare software			
2.2. Titularul disciplinei/a activităţilor de curs				Ş.l. dr. ing. Szántó Zoltán			
2.3. Titularul disciplinei/a activităţilor de seminar/laborator				Ş.l. dr. ing. Szántó Zoltán			
2.4. Anul de studiu	4	2.5. Semestrul	8	2.6. Tipul de evaluare	Ex	2.7. Regimul disciplinei	DI

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

3.1. Număr de ore pe săptămână	4	din care: 3.2. curs	2	3.3. laborator	2
3.4. Total ore din planul de învăţământ	56	din care: 3.5. curs	28	3.6. laborator	28
Distribuţia fondului de timp:					ore
a) Studiul după manual, suport de curs, bibliografie şi notiţe					21
b) Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren					24
c) Pregătire laboratoare, teme, referate					13
d) Tutoriat					5
e) Examinări					6
f) Alte activităţi:					
3.7. Total ore studiu individual					69
3.8. Total ore pe semestru					125
3.9. Numărul de puncte de credit					5

4. Precondiţii (acolo unde este cazul)

4.1. De curriculum	Programarea calculatoarelor, programare orientată pe obiecte, inginerie software
4.2. De competenţe	Cunoştinţe de programare, inginerie software, fundamente de testare

5. Condiţii (acolo unde este cazul)

5.1. De desfăşurare a cursului	
5.2. De desfăşurare a laboratorului	

6. Competențele specifice acumulate

Competențe profesionale	C3 - Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor C3.1 - Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice C3.2 - Utilizarea de cunoștințe interdisciplinare, a tiparelor de soluții și a uneltelor, efectuarea de experimente și interpretarea rezultatelor lor C3.3 - Aplicarea tiparelor de soluții cu ajutorul uneltelor și metodelor inginerești C3.4 - Evaluarea comparativă, inclusiv experimentală, a alternativelor de rezolvare, pentru optimizarea performanțelor C3.5 - Dezvoltarea și implementarea de soluții informatice pentru probleme concrete
Competențe transversale	

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1. Obiectivul general al disciplinei	• Familiarizarea viitorului specialist în calculatoare cu procedee de diagnozare și testare a sistemelor de calcul și/sau sistemelor similare
7.2. Obiectivele specifice	• Cunoașterea unor tehnici de testare, black-box și white-box, și proiectarea testelor pe baza lor • Înțelegerea noțiunii de analiză statică și aplicabilitatea ei complementar cu tehnicile de testare • Cunoașterea unor tehnici de testare specifice (orientate pe obiecte, pentru interfețe grafice, aplicații web, etc.) • Abilitatea de a planifica activitatea de testare și de a lua în considerare opțiunile de automatizare

8. Conținuturi

8.1 Curs	Număr de ore
1. Introducere în testare. Calitățile unui bun caz de test	2
2. Testarea black-box. Partiționarea în clase de echivalență	2
3. Testarea white-box. Criterii de acoperire cu teste	2
4. Semantica și verificarea formală a programelor	2
5. Analiză statică. Analiza fluxului de date	2
6. Verificarea programelor prin model checking	2
7. Specificare formală. Anotări și contracte software	2
8. Verificarea programelor concurente	2
9. Testarea programelor orientate pe obiecte	2
10. Testarea bazată pe modele	2
11. Automatizarea testării. Execuția simbolică	2
12. Testarea de integrare. Stabilirea unui plan de test	2
13. Testarea securității software	2
14. Recapitulare. Perspective în testarea și verificarea software	2

Bibliografie

Software Engineering (10th Edition) – I. Sommerville, Pearson, 2015

1. Software Architecture in Practice (3rd Edition) – Len Bass, Paul Clements, Rick Kazman, Addison-Wesley Professional, 2012
2. Software Engineering: A Practitioner's Approach (8th Edition) – Roger Pressman, Bruce Maxim, McGraw-Hill Education, 2014
3. Software Engineering Reviews and Audits – Boyd L. Summers, Auerbach, 2011
4. The Art of Software Testing (3rd Edition) - Glenford J. Myers, Corey Sandler, Tom Badgett, Wiley, 2011
5. Software Testing (2nd Edition) - Ron Patton, Sams Publishing, 2005
6. Software Testing: A Craftsman's Approach (4th Edition) - Paul C. Jorgensen, Auerbach Publications, 2013
7. Test Driven Development: By Example, Kent Beck, Addison-Wesley Professional; 1 edition (November 18, 2002)

8.2 Seminar/Laborator	Număr de ore
1. Introducere. Testarea folosind cadrul jUnit	4
2. Evaluarea acoperirii cu teste	2
3. Mock unit testing	2
4. Analiza statică. Detecția erorilor cu FindBugs	2
5. Analiza statică și verificarea specificațiilor	2
6. Analiza dinamică și profilarea execuției	2
7. Testarea interfețelor grafice	2
8. Testarea aplicațiilor web	4
9. Integrarea tehnicilor de testare. Testarea unui proiect realizat	8

Bibliografie

1. Software Engineering (10th Edition) – I. Sommerville, Pearson, 2015
2. Software Architecture in Practice (3rd Edition) – Len Bass, Paul Clements, Rick Kazman, Addison-Wesley Professional, 2012
3. Software Engineering: A Practitioner's Approach (8th Edition) – Roger Pressman, Bruce Maxim, McGraw-Hill Education, 2014
4. Software Engineering Reviews and Audits – Boyd L. Summers, Auerbach, 2011
5. The Art of Software Testing (3rd Edition) - Glenford J. Myers, Corey Sandler, Tom Badgett, Wiley, 2011
6. Software Testing (2nd Edition) - Ron Patton, Sams Publishing, 2005
7. Software Testing: A Craftsman's Approach (4th Edition) - Paul C. Jorgensen, Auerbach Publications, 2013
8. Test Driven Development: By Example, Kent Beck, Addison-Wesley Professional; 1 edition (November 18, 2002)

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

Specialiști cu cunoștințe privind testarea software sunt necesare la orice firmă de dezvoltare software, conform discuțiilor informale cu reprezentanți din industrie. Conținutul disciplinei a

fost corelat cu disciplina *Verificare si validare software*, Universitatea Politehnica Timișoara (http://staff.cs.upt.ro/~marius/curs/vvs/index.html, plan de învățământ http://staff.cs.upt.ro/~chirila/facultate/planuri/planuri/licenta/ctiro/2018-2019_AC_PI_CTI_Anii_III-IV.pdf , accesat 09.11.2019.).

10. Evaluare

Tip activitate		10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs		Cunoașterea noțiunilor de baza	Examinare scrisa	25%
10.5	Laborator	Rezolvarea unor probleme	Examinare scrisa	25%
		Exerciții practice de testare	Exerciții de testare la laborator	25%
		Integrarea tehnicilor de testare	Testarea unui proiect realizat de student	25%
10.6. Standard minim de performanță				
• Abilitatea de a proiecta teste după criterii black-box și white-box • Înțelegerea diferențelor dintre testare și analiză statică, și a contextului lor de aplicabilitate • Cunoașterea tehnicilor de testare specifice anumitor domenii (orientate pe obiecte, interfețe grafice, aplicații web)				

Semnătura titularului disciplinei:
Ș.l. dr. ing. Szántó Zoltán

Semnătura titularului /rilor de aplicații:
Ș.l. dr. ing. Szántó Zoltán

Data:
16. 09.
2019

Semnătura directorului de departament:
Conf. dr. ing. Domokos József

Semnătura coordonatorului programului de studii:
Ș.l.dr.ing. Szabó László Zsolt