

FIȘA DISCIPLINEI

Verificarea și validarea sistemelor software

Anul universitar 2025-26

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai din Cluj-Napoca
1.2. Facultatea	Facultatea de Matematică și Informatică
1.3. Departamentul	Departamentul de Matematică și Informatică al Liniei Maghiare
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Informatică
1.7. Forma de învățământ	cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei			Szoftverrendszerek verifikációja és validációja Verificarea și validarea sistemelor soft Verification and Validation of Software Systems				Codul disciplinei	MLM5014
2.2. Titularul activităților de curs			Lect. dr. ing. KOLUMBÁN Sándor					
2.3. Titularul activităților de seminar			Lect. dr. ing. KOLUMBÁN Sándor					
2.4. Anul de studiu	3	2.5. Semestrul	6	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	DS	

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

3.1. Număr de ore pe săptămână	5	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	3
3.4. Total ore din planul de învățământ	60	din care: 3.5. curs	24	3.6 seminar/laborator/proiect	36
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					25
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					35
Tutoriat (consiliere profesională)					7
Examinări					3
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				90	
3.8. Total ore pe semestru				150	
3.9. Numărul de credite				6	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	- MLM5006 Programare orientată obiect/Objektumorientált programozás/Object Oriented Programming - MLM5008 Metode avansate de programare/Haladó szintű programozási módszerek/Advanced Programming Methods
4.2. de competențe	

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sala de curs dotată cu tablă și proiector video
5.2. de desfășurare a seminarului/ laboratorului	Sală de seminar dotată cu tablă și proiector video

6.1. Competențele specifice acumulate¹

Competențe profesionale/ esențiale	• C2.1 Identificarea metodologiilor adecvate pentru dezvoltarea sistemelor software • C2.2 Identificarea și explicarea mecanismelor adecvate pentru specificarea sistemelor software • C2.3 Utilizarea metodologiilor, mecanismelor de specificare și mediilor de dezvoltare pentru crearea de aplicații informatice • C2.4 Utilizarea criteriilor și metodelor adecvate pentru evaluarea aplicațiilor informatice
Competențe transversale	• CT1 Aplicarea regulilor de organizare și eficientizare a muncii și a unei atitudini responsabile în domeniul didactic-științific, în vederea valorificării creative a propriului potențial, respectând principiile și normele deontologice profesionale • CT3 Utilizarea metodelor și tehnicilor eficiente pentru învățare, colectarea de informații, cercetare și dezvoltarea capacităților de dobândire a cunoștințelor, satisfacerea nevoilor unei societăți dinamice, comunicarea în limba română și într-o limbă internațională

6.2. Rezultatele învățării

Cunoștințe	• Conceptele de bază ale verificării și validării software-ului • Cadrul de integrare al modelului de maturitate a capacității • Locul Ver. Val. în ciclul de viață al dezvoltării software-ului • Modele de asigurare a calității software-ului (McCall, Bohm, FURPS) • Teste statice, complexitate ciclică • Virtualizare și sisteme de compilare (Jenkins, TeamCity) • CI/CD • Automatizarea web (JSON Wire Protocol) • Testare mobilă (Appium) • TDD-BDD • Strategii de rollback
Aptitudini	• Testarea unităților și integrării componentelor software • Testarea finală a aplicațiilor desktop, web și mobile • Configurarea metodologiei de testare • Evaluarea eficacității și relevanței testelor
Responsabilități și autonomie	• Participarea la procese responsabile de dezvoltare software utilizând instrumente de verificare și validare software.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

7.1 Obiectivul general al disciplinei	Scopul cursului este de a prezenta fundamentele teoretice și abordările industriale ale asigurării calității software-ului, acoperind ciclul de viață al testării, diverse tehnici de testare și instrumente moderne pentru automatizare, CI/CD și virtualizare. Studenții vor fi capabili să dezvolte strategii de testare, să proiecteze și să execute teste unitare, de integrare și end-to-end pe diferite platforme (desktop, web, mobil) și să măsoare eficiența și eficacitatea testării. Scopul cursului este de a dezvolta competențe care să asigure pregătirea studenților pentru a participa la proiecte profesionale de dezvoltare de software și asigurare a calității, utilizând standarde și cadre industriale dovedite.
7.2 Obiectivele specifice	În cadrul cursului, vom utiliza cadrul .Net pentru a examina fluxurile de lucru de verificare și validare pentru aplicații desktop, web și mobile. Pregătirea rapoartelor de testare, testarea integrată în platformele CI/CD.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
Introducere - Istoria erorilor software - Asigurarea calității software-ului (SQA) - Caracteristici de calitate Teste unitare	Prelegere, prezentare, explicații, exemple	
Definiția verificării și validării - Clasificarea testelor - Integrarea modelului de maturitate a capacității Verificarea și validarea în ciclul de viață al dezvoltării software-ului	Prelegere, proiectie, explicații, exemple	
Modele de asigurare a calității software - McCall - Bohm FURPS	prelegere, proiectie, explicații, exemple	
Analiza statică a codului - Lexical - Sintactică Analiza fluxului de control	prelegere, proiectie, explicații, exemple	
Analiza statică a codului - Complexitate ciclică - Analiza fluxului de date - Testare bazată pe metrici Analiza securității	prezentare, proiectie, explicații, exemple	
Virtualizare - Container Dev/Ops	prelegere, proiectie, explicații, exemple	
CI/CD	prezentare, proiectie, explicații, exemple	
Automatizare web (Selenium)	prelegere, proiectie, explicații, exemple	
Suport pentru automatizare în aplicații	prelegere, proiectie, explicații, exemple	
Testare mobilă	prelegere, prezentare, explicații, exemple	
TDD-BDD Cucumber	prezentare, proiectie, explicații, exemple	
Revertire automată	prelegere, proiectie, explicații, exemple	
Introducere - Istoria erorilor software - Asigurarea calității software-ului (SQA) - Caracteristici de calitate	Prelegere, prezentare, explicații, exemple	

Teste unitare		
Definiția verificării și validării - Clasificarea testelor - Integrarea modelului de maturitate a capacității Verificarea și validarea în ciclul de viață al dezvoltării software-ului	Prelegere, proiecție, explicații, exemple	
Modele de asigurare a calității software - McCall - Bohm FURPS	prelegere, proiecție, explicații, exemple	
Bibliografie		
8.2 Seminar / laborator	Metode de predare	Observații
Teste unitare – Sintaxă fluentă	Rezolvarea sarcinilor	
Mocking și generarea automată de date	Rezolvarea sarcinilor	
Fluxuri de lucru pentru testarea statică și executarea automată a testelor dinamice în conductele CI.	Rezolvarea sarcinilor	
Automatizarea și testarea aplicațiilor web	Soluționarea sarcinilor	
Automatizarea și testarea aplicațiilor mobile	Soluționarea sarcinilor	
Dezvoltare/testare BDD	Rezolvarea sarcinilor	
Bibliografie		

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

- Conținutul disciplinei corespunde conținutului tradițional al logicii computaționale predate în marile universități.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoașterea conceptelor de bază	Test scurt la fiecare prezentare	10
	Bazele teoretice	Examen scris	60
10.5 Seminar/laborator	Cunoștințe despre subiectele de laborator	La fiecare seminar se susține un test scurt.	5
	Cunoștințe despre subiectele de laborator	Realizarea sarcinilor de laborator	25
10.6 Standard minim de performanță			
• Finalizarea a cel puțin 50% din activitățile de la jumătatea anului • Finalizarea a cel puțin 50% din examenele scrise			

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

	Nu se aplică.
--	---------------

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "Nu se aplică".

Data completării

2025.05.01

Semnătura titularului de curs

Lect. dr. ing. KOLUMBÁN Sándor

Semnătura titularului de seminar

Lect. dr. ing. KOLUMBÁN Sándor

Data avizării în departament

2025.05.01

Semnătura directorului de departament

Conf. dr. ANDRÁS Szilárd Károly