

Szoftvertervezés és -fejlesztés Intézet			Mintatanterv szerinti 6. félév 2026-27-1			
Tantárgy neve:	Kódja:	Kredit:	Óraszám			
				ea	tyg	lab
Modern szoftvertechnológia	NSXST3HBNF	4	heti	2		
Tárgyfelelős: Dr. Dineva Adrienn			Beosztás: egyetemi docens			
Oktató(k): Dr. habil Tick József						
Előtanulmányi feltételek:		NSXST1HBNF	Szoftvertechnológia			
Számonkérés módja:		Vizsga				
A tananyag						
Oktatási cél:	A tárgy célja, hogy megismertesse a hallgatókkal a professzionális nagyvállalati szoftverfejlesztés eszközeit és módszereit, nagy hangsúllyal a legjobb gyakorlatokra és a tervezés, a fejlesztési folyamat és az üzemeltetés során előkerülő kihívásokra. A hallgató gyakorlati ismereteket szerez egy magas funkcionális és nemfunkcionális követelményekkel rendelkező nagyvállalati alkalmazás fejlesztéséhez, teszteléséhez, üzemeltetéséhez és karbantartásához szükséges eszközökről és módszerekről. Bemutatjuk a legfontosabb keretrendszereket, ezek alkalmazását webes és más alkalmazások fejlesztése során. A teljes szoftveréletről lefedő technológiákat és technikákat mutatunk be, hogy a hallgató a kurzus elvégzése után magabiztosan tudjon mozogni egy nagyléptékű szoftverfejlesztési projektben.					
Tematika:	A tantárgy kereteiben bemutatjuk azokat a kihívásokat és a megoldásukat szolgáló legkorszerűbb technológiákat, amelyek a nagyvállalati szoftverfejlesztés során leggyakrabban előkerülnek. Egy-egy problémakör kapcsán megvizsgáljuk, hogy jellemzően milyen gyakorlati eszközök állnak rendelkezésre a problémák megoldására, melyek azok a szakmai elvek, amelyek megjelennek bennük, hogy ezzel a hallgató egyfelől képes legyen a nagyvállalati rendszerekben előforduló feladatok gyakorlati megoldására is, másfelől elsajátítsa azt a gondolkodásmódot, amivel a folyamatosan fejlődő és egyre újabb feladatokat adó nagyvállalati szoftverfejlesztésben helyt tud állni.					

Féléves ütemezés	
Oktatási hét (konzultáció)	Témakör
1.	Ipari szoftvertermelés
2.	SDLC – DevOps – DevSecOps
3.	Kódolás a gyakorlatban
4.	Verziókezelés és a Build folyamat
5.	Continuous Integration / Continuous delivery
6.	Verifikáció – validáció - tesztelés
7.	Konténerizáció, Orchestration
8.	Microservices
9.	Cloud technológiák és szolgáltatások
10.	Skálázható adattárolás és –feldolgozás
11.	Konfiguráció menedzsment
12.	Esettanulmány
13.	Féléves ZH
14.	PótZH
Félévközi követelmények	

Évközi jegy / aláírás megszerzésének feltételei:	A féléves ZH, illetve a PótZH eredményei alapján
Zárthelyi dolgozatok	
Oktatási hét	Témakör
13	Féléves ZH a félév teljes anyagából
14	PótZH sikertelen féléves ZH esetén
Az évközi jegy kialakításának módszere (csak évközi jegyes tárgyak esetében töltendő ki)	
	0-19% letiltva
	20-49% aláírás megtagadva, pótolható
	50-100% aláírás
Pótlás módja	
A ZH / évközi jegy / aláírás pótlásának módja:	Abban az esetben, ha a Féléves ZH eredménye nem éri el az 50%-ot, akkor a hallgató a dolgozatot PótZH formájában pótolhatja. Ha a hallgató a szorgalmi időszakban nem szerezte meg az aláírást (50%), de a Féléves ZH-n, vagy PótZH-n legalább 20%-ot elért, akkor a vizsgaidőszak elején, a TVSZ-ben foglaltak szerint az aláírást egy alkalommal pótolhatja
Vizsga módja (csak vizsgás tantárgy esetében töltendő ki)	
Írásbeli vizsga	
Vizsgajegy kialakítása (csak vizsgás tantárgy esetében töltendő ki)	
Sikeres aláírás esetén a vizsgaidőszakban teljesített vizsga adja a félévzáró érdemjegyet.	
Az egyes érdemjegyek ponthatárai:	
	0-49% Elégtelen
	50-62% Elégséges
	63-74% Közepes
	75-87% Jó
	88-100% Jeles
Irodalom	
Kötelező:	A moodle rendszerbe feltöltött anyagok
Ajánlott:	Ian Sommerville: Engineering Software Products, Pearson Education, Inc. 2020. Jez Humble, David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley Professional, 2010 Kenneth Geisshirt, Emanuele Zattin, Aske Olsson, Rasmus Voss, "Git Version Control Cookbook: Leverage version control to transform your development workflow and boost productivity", Packt Publishing, 2018 Bahaaldine Azarmi, "Scalable Big Data Architecture: A practitioners guide to choosing relevant Big Data architecture" Springer, 2016