

Szoftvertervezés és -fejlesztés Intézet			Mintatanterv szerinti 4. félév 2024-25-2			
Tantárgy neve:	Kódja:	Kredit:	Óraszám			
				ea	tyg	lab
Nagy rendszerek fejlesztésének technológiája	NSXNR1HBNF	4	heti	3		
Tárgyfelelős: Prof. Dr. Vámosy Zoltán			Beosztás: egyetemi tanár			
Oktató(k): Dr. habil Tick József (2/3) Kovács András (1/3)						
Előtanulmányi feltételek:		NSXST1HBNF	Szoftvertechnológia			
Számonkérés módja:		Évközi jegy				
A tananyag						
Oktatási cél:	A tárgy keretében a hallgatók elsajátítják a nagy szoftver rendszerek fejlesztésének lehetőségeit. Megismerkednek a komplex szoftver rendszerek modell alapú fejlesztésével, a szoftvertechnológia haladó elveivel, paradigmáival és azok alkalmazási lehetőségeivel.					
Tematika:	A szoftver mint termék jellemzői, nagy rendszerek fejlesztésének speciális kihívásai, nagy rendszerek gyártásának projekt menedzselése – koncepciók, projekt és folyamat metrikák, becslés, ütemezés, menedzsment, minőség menedzsment, Software Process Improvement, CMMI, szoftver rendszerek biztonsági kérdései – Sérülékenységi fajták, hatékony védekezés, biztonságos kód, MS módszertan, konfiguráció kezelés, folyamatos integráció – verziócontrol rendszerek, branching, work item tracking, projekt átadás és rendszerintegráció – Scaled Agile Framework (SAFe modell), DevOps mindset, Domain modell tulajdonságai, szerepe a szoftvertervezésben, DDD (Domain Driven Design), szoftver architektúra és Systematic Architecture Design, Risk based testing, Teszt stratégia kialakítása a fejlesztés, a tesztelés az ipari gyakorlatban. Design patternek					

Féléves ütemezés	
Oktatási hét (konzultáció)	Témakör
1.	A szoftver mint termék / Creational patterns I. (Factory method, Abstract factory, builder)
2.	Nagy rendszerek fejlesztésének, speciális kihívásai
3.	Szoftver minőség / Creational patterns II. (Singleton, Prototype)
4.	Szoftverfolyamat továbbfejlesztés CMMI
5.	Biztonságos kód / Structural patterns I. (Adapter, Bridge, Composite, Flyweight)
6.	Biztonságos módszertanok
7.	Konfiguráció kezelés, folyamatos integráció / Structural patterns II. (Facade, Proxy, Decorator)
8.	Domain modell
9.	Szisztematikus architektúra tervezés / Behavioral patterns I. (Iterator, Chain of responsibility, Visitor, Observer, Command, Mediator)
10.	Tesztelés az ipari gyakorlatban
11.	Rizikó alapú tesztelés / Behavioral patterns II. (Strategy, Template method, Memento, State, Interpreter)
12.	Projektátadás és integráció
13.	Féléves ZH
14.	PótZH
Félévközi követelmények	

Évközi jegy / aláírás megszerzésének feltételei:	A féléves ZH-k, illetve a PótZH-k eredményei alapján. A zárthelyi dolgozat két részből áll, amelyet a hallgatók külön időpontban írnak meg.
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% elégtelen (1) 50-62% elégséges (2) 63-74% közepes (3) 75-87% jó (4) 88-100% jeles (5)
Pótlás módja	
A ZH / évközi jegy / aláírás pótlásának módja:	Abban az esetben, ha a ZH nem éri el az 50%-ot, akkor a hallgató a dolgozatot PótZH formájában pótolhatja a 14. héten. Ha a hallgató a szorgalmi időszakban nem szerezte meg az évközi jegyet (min. 50%), de a ZH-n, vagy PótZH-n legalább 20%-ot elért, akkor a vizsgaidőszak elején, a TVSZ-ben foglaltak szerint az évközi jegyet egy alkalommal pótolhatja.
Vizsga módja (csak vizsgás tantárgy esetében töltendő ki)	
Vizsgajegy kialakítása (csak vizsgás tantárgy esetében töltendő ki)	
Az egyes érdemjegyek ponthatárai:	
Irodalom	
Kötelező:	A moodle rendszerbe feltöltött oktatási anyagok
Ajánlott:	Ian Sommerville: Software Engineering, 10th edition, Addison-Wesley, 2015 Roger S. Pressman, Bruce R. Maxim: Software Engineering – A Practitioner’s Approach 8th edition McGraw-Hill, 2015 Bruce R. Maxim and Roger S. Pressman: Software Engineering: A Practitioner’s Approach, 8/e McGraw Hill, 2019 Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns: Elements of Reusable Object Oriented Software, Addison Wesley, 1994 Martin Fowler: Refactoring, Addison Wesley, 1999