

<b>Óbudai Egyetem</b> Neumann János Informatikai Kar		Szoftvertervezés és -fejlesztés Intézet			
<b>Tantárgy neve és kódja:</b> Nagy rendszerek fejlesztésének technológiája (NIXNR1LBEE)		<b>Kreditérték:</b> 3			
<i>Gazdaságinformatikus BSc szak</i>		<i>Esti tagozat 2022/23 tanév II. félév</i>			
Tantárgy oktató(i): Dr. habil. Tick József					
Előtanulmányi feltételek: (kóddal)					
Heti óraszámok:		Előadás: 1	Tantermi gyak.: 0	Laborgyakorlat: 0	Konzultáció: 0
Számonkérés módja:		Évközi jegy			
<b>A tananyag</b>					
<p><i>Oktatási cél:</i> 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. Ismereteket szereznek a minőségszemléletű szoftverfejlesztési folyamatról, a szoftver teszteléstől, a verifikációról és validációról, valamint a modern agilis szoftverfejlesztési módszerek gyakorlatban történő hatékony alkalmazásáról.</p> <p><i>Tematika:</i> A szoftver mint termék jellemzői, szoftver típusok, a szoftver fejlesztési folyamat - problémák, elvek, különböző megközelítések, azok jellemzői, a szoftver projekt menedzselése – koncepciók, projekt és folyamat metrikák, becslés, ütemezés, menedzsment, minőség menedzsment – koncepciók, technikák, szabványok, megoldások, Software Process Improvement, CMMI, szoftver rendszerek biztonsági kérdései – Sérülékenységi fajták, hatékony védekezés, biztonságos kód, MS módszertan, konfiguráció kezelés, folyamatos integráció – verziócontrol rendszerek, build rendszerek, branching, work item tracking, felhő-alapú fejlesztési alapok, projekt átadás és rendszerintegráció – Scaled Agile Framework (SAFe modell), DevOps mindset, Domain modell – Domain modell tulajdonságai, szerepe a szoftvertervezésben, DDD (Domain Driven Design) alapok, szoftver architektúra és Systematic Architecture Design – Szoftver architektúra tervezésének lépései, probléma- és megoldástér, követelmények, design koncepció, walking skeleton, Risk based testing, Refactoring – Teszt stratégia kialakítása a fejlesztés előtt, user story fejlesztése rizikók mentén, szoftver karbantartása, technikai adósság fogalma, kód metrikák és indikátorok, kód refactoring módszerek.</p>					

<b>Féléves ütemezés</b>					
Oktatási hét (konzultáció)	Témakör				
1	A szoftver, mint termék sajátosságai				
2	Szoftver folyamat modellek				
3	Agilis SW fejlesztési módszertanok				
4	A szoftverfejlesztés minőségi kérdései				
5	Konfiguráció kezelés és folyamatos integráció				
6	Rektori szünet				
7	Biztonságos kód - az MS-SDL módszertan				
8	Domain modell				
9	Szisztematikus architektúra tervezés				
10	Tesztelés az ipari gyakorlatban				
11	Rizikó-alapú tesztelés				
12	Projektátadás és rendszerintegráció				
13	Féléves ZH				
14	Pót ZH				
<b>Félévközi követelmények</b>					
Évközi jegy / aláírás megszerzésének feltételei: a ZH, illetve a PótZH eredményei alapján.					
<table border="1"> <tr> <td>0-19%</td> <td>letiltva</td> </tr> <tr> <td>20-100%</td> <td>aláírás</td> </tr> </table>		0-19%	letiltva	20-100%	aláírás
0-19%	letiltva				
20-100%	aláírás				
<b>Zárthelyi dolgozatok</b>					
Oktatási hét (konzultáció)	Témakör				
13	Féléves ZH				
14	PótZH				
<b>A félévzáró érdemjegy (é) kialakításának módszere</b>					
Az évközi jegy / aláírás kialakításának módja: Az egyes érdemjegyek ponthatarai:					

0-49%	elégtelen (1)
50-62%	elégséges (2)
63-74%	közepes (3)
75-87%	jó (4)
88-100%	jeles (5)

#### **Pótlás módja**

Az 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. Ebben az esetben az évközi jegy meghatározása során a pótzh eredményét kell figyelembe venni. Ha a hallgató a szorgalmi időszakban nem szerezte meg az évközi jegyet, 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**

#### **Vizsgajegy kialakítása**

#### **Irodalom**

##### **Kötelező:**

Ian Sommerville: Software Engineering, 10th edition, Addison-Wesley, 2015

##### **Ajánlott:**

Roger S. Pressman, Bruce R. Maxim: Software Engineering – A Practitioer’s Approach 8th edition McGraw-Hill, 2015.

##### **Egyéb segédletek:**

A moodle rendszerbe feltöltött oktatási anyagok