

A TANTÁRGY ADATLAPJA

1. A képzési program adatai

1.1. Felsőoktatási intézmény	Babeş–Bolyai Tudományegyetem
1.2. Kar	Fizika
1.3. Intézet	Magyar fizika intézet
1.4. Szakterület	Fizika
1.5. Képzési szint	Alapképzés
1.6. Szak / Képesítés	Fizika informatika

2. A tantárgy adatai

2.1. A tantárgy neve	Szoftvertchnológiák						
A tantárgy kódja	MLM5011						
2.2. Az előadásért felelős tanár neve	Dr.-Ing. Barabás László						
2.3. A szemináriumért felelős tanár neve	Dr. Szenkovits Annamária						
2.4. Tanulmányi év	3	2.5. Félév	5	2.6. Értékelés módja	vizsga	2.7. Tantárgy típusa	választható - alap

3. Teljes becsült idő (az oktatási tevékenység féléves óraszama)

3.1. Heti óraszám	4	3.2. amelyből: előadás	2	3.3. szeminárium/labor	2
3.4. Tantervben szereplő össz-óraszám	56	3.5. amelyből: előadás	28	3.6. szeminárium/labor	28
A tanulmányi idő elosztása:					óra
A tankönyv, a jegyzet, a szakirodalom vagy saját jegyzetek tanulmányozása					10
Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás					14
Szemináriumok / laborok, házi feladatok, portfóliók kidolgozása					14
Egyéni készségfejlesztés (tutorálás)					4
Vizsgák					2
Más tevékenységek:					
3.7. Egyéni munka össz-óraszama					44
3.8. A félév össz-óraszama					100
3.9. Kreditszám					4

4. Előfeltételek (ha vannak)

4.1. Tantervi	• nincs
4.2. Kompetenciabeli	• nincs

5. Feltételek (ha vannak)

5.1. Az előadás lebonyolításának feltételei	• Projektorral ellátott előadóterem
5.2. A szeminárium / labor lebonyolításának feltételei	• Számítógépes terem, verziókövető rendszerrel, software tervező rendszerrel, hibák kezelését biztosító rendszerrel, software fejlesztői rendszerrel ellátott terem

6. Elsajátítandó jellemző kompetenciák

Szakmai kompetenciák	<p>C2 Informatikai alkalmazások fejlesztése és karbantartása</p> <p>C2.1 A szoftverrendszerek megfelelő fejlesztési módszereinek beazonosítása</p> <p>C2.2 Szoftverrendszerek specifikációjának megfelelő mechanizmusok beazonosítása és magyarázata</p> <p>C2.3 Módszerek, specifikációs mechanizmusok és fejlesztési környezetek alkalmazása az informatikai alkalmazások fejlesztéséhez</p> <p>C2.4 Megfelelő kritériumok és módszerek használata az alkalmazások értékeléséhez</p> <p>C2.5 Dedikált informatikai projektek megvalósítása</p>
Transzverzális kompetenciák	<p>CT1 A szervezett és hatékony munka szabályainak, a didaktikai-tudományos területhez való felelősségteljes hozzáállás alkalmazása a saját potenciál kreatív értékesítéséhez, a szakmai etika alapelveinek és normáinak tiszteletben tartásával</p> <p>CT2 Interdiszciplináris csoportban szervezett tevékenységek hatékony lebonyolítása és az interperszonális kommunikáció, a különféle csoportokhoz való viszony és együttműködés empátikus képességének fejlesztése</p> <p>CT3 Hatékony módszerek és technikák használata tanulásra, információszerezésre, kutatásra és a tudásszerzési kapacitások fejlesztésére, egy dinamikus társadalom igényeinek való megfelelésre, román és egy nemzetközi nyelven történő kommunikációra</p>

7. A tantárgy célkitűzései (az elsajátítandó jellemző kompetenciák alapján)

7.1. A tantárgy általános célkitűzése	Szolid mérnöki elveknek az alkalmazása , azzal a céllal, hogy gazdaságossági alapon software-t létrehozzunk, amelyek, megbízhatóak és valóságos környezetben futnak.
7.2. A tantárgy sajátos célkitűzései	A software követelményeinek kezelése, software tervezés, software ellenőrzés során szükséges elméleti és gyakorlati ismeretek, képességek elsajátítása és ezek tevékenységek, folyamatok software eszközökkel való támogatása.

8. A tantárgy tartalma

8.1 Előadás	Didaktikai módszerek	Megjegyzések
<p>1. Software technológia bevezetése</p> <ul style="list-style-type: none"> • Software technológia történeti háttere • Software technológia kihívásai • Software technológia, mint tudomány • Software fejlesztés életciklusa 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
<p>2 Software követelmények</p> <ul style="list-style-type: none"> • Software követelmények meghatározása • Software követelmények osztályozása • Software követelmény folyamatok • Software követelmény meghatározási technikák • Software követelmény validálás 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
<p>3 Software architektúra</p> <ul style="list-style-type: none"> • Software architektúra meghatározása • Software architektúrát alkotó elemek meghatározása • Software architektúra folyamatok • Software architektúra ismertebb típusainak és 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás 	

<p>mintáinak ismertetése</p> <ul style="list-style-type: none"> • Software architektúra nem azonos a software design-al (tervezéssel) 	<ul style="list-style-type: none"> - közös megbeszélés - magyarázat- 	
<p>4 Software architektúra</p> <ul style="list-style-type: none"> • Software architektúra meghatározása • Software architektúrát alkotó elemek meghatározása • Software architektúra folyamatok • Software architektúra ismertebb típusainak és mintáinak ismertetése • Software architektúra nem azonos a software design-al (tervezéssel) 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
<p>5 Software tervezés</p> <ul style="list-style-type: none"> • Objektum orientált analízis • Software tervezési modell dinamikus elemei 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
<p>6. Software tervezés adatmodellje és kommunikációs modelljei</p> <ul style="list-style-type: none"> • Software tervezési modell komponens elemei • Software tervezési modell telepítés elemei • Software tervezési kommunikációs modell elemei 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
<p>7. Software implementálás</p> <ul style="list-style-type: none"> • Implementálás előírásai • Formai követelmények • Fejlesztési prioritások • Review • Statikus és dinamikus ellenőrzés 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
<p>8. Software tesztelés</p> <ul style="list-style-type: none"> • Software tesztelés meghatározása • Software tesztelési módszerek • Tesztspecifikáció és teszteredmény • Hiba és változás követő rendszerek • Verifikáció nem azonos a validációval 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
<p>9. Software fejlesztési folyamatok</p> <ul style="list-style-type: none"> • Tevékenység és folyamat fogalmak meghatározása • Ismertebb folyamatok ismertetése • Vizes és társai folyamatok ismertetése • Inkrementáló folyamatok ismertetése • Iteratív folyamatok ismertetése 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
<p>10. Agilis software fejlesztési folyamatok</p>	<ul style="list-style-type: none"> - rövid ismétlés 	

<ul style="list-style-type: none"> • Agilis software folyamat meghatározása • Folyamat orientált versus Változás és ember centrikus folyamat • Inkrementáló és iteratív folyamat • SCRUM ismertetése • XP ismertetése • Tervezés és Agilis folyamat kapcsolata 	<ul style="list-style-type: none"> - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
11 Projekt fogalma <ul style="list-style-type: none"> • Projekt fogalmának a meghatározása • Projekt jellemzői • Projekt típusok • Projekt életciklusai • Projekt folyamatcsoportok • Projekttervezés és ütemezés 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
12 Konfigurációs rendszerek <ul style="list-style-type: none"> • Konfiguráció management meghatározása • Verziókövetés fogalma • Subvesion és ClearCase ismertetése 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
13. Minőség biztosítás <ul style="list-style-type: none"> • Minőség biztosítási rendszer meghatározása • ISO 9001 rendszer ismertetése • Fő és támogató folyamatok • CMMI ismertetése 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
14 IT Projekt management <ul style="list-style-type: none"> • Projekt management meghatározása • Szervezetek és tervezésük • Embertípusok és csapatépítés • Motivációs elméletek • Rizikó management • Konfliktuskezelés • Vezetési stílusok 	<ul style="list-style-type: none"> - rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat- 	
Könyvészet <ol style="list-style-type: none"> 1. FLAATEN, P.O., MCCUBBREY, D.J., O'RIORDAN, P.D., BURGESS, K.: Foundations of Business Systems, Dryden Press, 1st ed. 1989, 2nd ed. 1997. 2. FOWLER, M., SCOTT, K.: UML Distilled: A Brief Guide to the Standard Object Modeling Language, Addison-Wesley, 2nd ed., 1999. 3. IACOBSON, I., BOOCH, G., RUMBAUGH, J.: The Unified Software Development Process, Addison-Wesley, 1999. 4. R.C. MARTIN: Agile Software Development: Principles, Patterns, and Practices, Prentice Hall, 2002. 5. W. ZUSER, S. BIFFL: Th. GRECHENING, M. KÖHLE Software Engineering, Pearson Studium 2004 6. D. A. GUSTAFSON: Theory and Problems of Software Engineering, McGraw-Hill, 2002 7. R. S. PRESSMAN: Software Engineering A practitioner's approach, McGraw-Hill, 2001 8. D. KOVÁCS: Lehel Rendszerek elemzése és tervezése, Egyetemi Kiadó, Kolozsvár, 2004 9 B. PÂRV: Analiza si proiectarea programelor, Alba Iulia Universitatea "1 Decembrie 1918", 2003 		

10. I. SOMMERVILLE: Software Engineering, Addison-Wesley, 5th ed. 1996, 6th ed. 2000, 7th ed. 2004.
11. Agile Modeling Homepage, [<http://www.agilemodeling.com>].
12. Software Engineering Body of Knowledge, IEEE, 2004. [<http://www.swebok.org>].
13. SysML specification <http://www.sysml.org/specs/>
14. M. COHN: Agile Estimating and Planning. Pearson Education (2007).
15. K. BECK: Test Driven Development: By Example, Addison-Wesley Professional (2002)
16. R. OSHEROVE: The Art of Unit Testing: Manning Publications (2009).
17. G.J MYERS: The Art of Software Testing: John Wiley & Sons, Inc.(2012).
18. M.FOWLER: Patterns of Enterprise Application Architecture, Addison-Wesley(2009).

8.2. Szeminárium / Labor	Didaktikai módszerek	Megjegyzések
1. Esettanulmány: követelmények meghatározása,	<ul style="list-style-type: none"> - a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése 	
2. Dokumentumkezelés	<ul style="list-style-type: none"> - a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése 	
3. Esettanulmány: architektúra meghatározása	<ul style="list-style-type: none"> - a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése 	
4. Verziókövető eszköz használata	<ul style="list-style-type: none"> - a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, 	

	megtárgyalása, értelmezése	
5. UML nyelvezet, UML modellezés	<ul style="list-style-type: none"> - a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése 	
6. Modellező eszköz használata	<ul style="list-style-type: none"> - a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése 	
7. Esettanulmány: Software tervezés UML segítségével	<ul style="list-style-type: none"> - a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése 	
8. UML modell megalkotása, statikus elemek	<ul style="list-style-type: none"> - a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése 	
9. Projekttervezés, ütemezés	<ul style="list-style-type: none"> - a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak 	

	- az eredmények közös ellenőrzése, megtárgyalása, értelmezése	
10. UML modell megalkotása, dinamikus elemek	- a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése	
11. Esettanulmány: Tesztelés megtervezése	- a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése	
12. Hiba és változás kezelő rendszer használata	- a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése	
13. Esettanulmány: Kiértékelés	- a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése	
14. Projekttervező eszköz használata	- a hosszú távú projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése	

	- a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése	
--	---	--

9. Az episztemikus közösségek képviselői, a szakmai egyesületek és a szakterület reprezentatív munkáltatói elvárásainak összhangba hozása a tantárgy tartalmával.

- A hosszú távú projekt és szakmai gyakorlatok során kikristályosodott elvárásoknak megfelelően, az oktatott tantárgyba beépítettnek konkrét, IT ipari példák, amelyek segítenek a hallgatóknak abban, hogy ne csak elméleti tudást sajátítsanak el, hanem összefüggéseiben és konkrétan lássák azokat a problémákat, amelyekkel az IT iparban nap, mint nap küszködnek az IT-ek.

10. Értékelés

Tevékenység típusa	10.1 Értékelési kritériumok	10.2 Értékelési módszerek	10.3 Aránya a végső jegyben
10.4 Előadás / Labor	Alapfogalmak, alap software technológia fogalmak ismerete és használata. Laborfeladatok helyessége	Minden héten történő előadásanyaghoz fűződő írásbeli ellenőrzés (40%) Hétről hétre helyesen implementált és személyesen bemutatott, határidőre kitűzött laborfeladatok ellenőrzése. (30%)	70%
10.5 Előadás / Labor	Alapfogalmak, alap software technológia fogalmak ismerete és használata	Félév végi írásbeli vizsga elméleti jellegű feladatokból.	30%
10.6 A teljesítmény minimumkövetelményei			
<ul style="list-style-type: none"> • Minden héten történő előadásanyaghoz fűződő írásbeli ellenőrzés • Összes kitűzött laborfeladat határidőre való megoldása • Legalább 5-s minősítés elérése az írásbeli vizsgán 			

Kitöltés dátuma

2018.04.02.

Előadás felelőse

Dr. Barabás László

Szeminárium felelőse

Dr. Szenkovits Annamária

Az intézeti jóváhagyás dátuma

2018.04.02.

Intézetigazgató,

dr. Járai-Szabó Ferenc egyet. docens