

A TANTÁRGY ADATLAPJA

1. A képzési program adatai

1.1. Felsőoktatási intézmény	Babeş–Bolyai Tudományegyetem
1.2. Kar	Matematika és Informatika
1.3. Intézet	Magyar Matematika és Informatika
1.4. Szakterület	Informatika
1.5. Képzési szint	Alapképzés
1.6. Szak / Képesítés	Informatika és Matematika-informatika

2. A tantárgy adatai

2.1. A tantárgy neve	Software technológia						
A tantárgy kódja	MLM5011						
2.2. Az előadásért felelős tanár neve	Dr.-Ing. Barabás László						
2.3. A szemináriumért felelős tanár neve	Dr.-Ing. Barabás László						
2.4. Tanulmányi év	2	2.5. Félév	2	2.6. Értékelés módja	vizsga	2.7. Tantárgy típusa	kötelező - 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					28
Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás					20
Szemináriumok / laborok, házi feladatok, portofóliók kidolgozása					40
Egyéni készségfejlesztés (tutorálás)					4
Vizsgák					2
Más tevékenységek:					
3.7. Egyéni munka össz-óraszama					90
3.8. A félév össz-óraszama					146
3.9. Kreditszám					6

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, verzió követő rendszerrel, software tervező rendszerrel, hibák kezelését biztosító rendszerrel, software fejlesztői rendszerrel ellátott terem

6. Elsajátítandó jellemző kompetenciák

Szakmai kompetenciák	<ul style="list-style-type: none"> • A software technológia alapjait képező törvényszerűségek, módszertanok, -ajánlások ismerete és ezek alkalmazása • A software technológiából elsajátított ismeretek alkalmazása tipikusan software problémák megoldásában • A software technológiák kritikus elemzésének készsége és az ismeretek alkalmazása a konkrét software életciklus esetében
Transzverzális kompetenciák	<ul style="list-style-type: none"> • A feladatok teljesítése meghatározott időn belül, maximális komolysággal és felelősségérzettel • A csapatmunkára való képesség • Színvonalas dokumentációs munkára való képesség • Tervezésre és ellenőrzésre való igény és hajlam

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 reális 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
1. Hét: SOFTWARE TECHNOLÓGIA BEVEZETÉS Software technológia történeti háttere Software technológia kihívásai Software technológia, mint tudomány Software fejlesztés életciklusa	- rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
2 Hét: SOFTWARE KÖVETELMÉNYEK Software követelmények definíciója Software követelmények osztályozása Software követelmény folyamatok Software követelmény meghatározási technikák Software követelmény validálás	- rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
3 Hét: SOFTWARE ARCHITEKTÚRA 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)	- rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
4 Hét: SOFTWARE ARCHITEKTÚRA Software architektúra meghatározása	- rövid ismétlés - a hallgatók aktív	

Software architektúrát alkotó elemek meghatározása Software architektúra folyamatok Software architektúra ismertebb típusainak és mintáinak ismertetése Software architektúra nem azonos a software design-al (tervezéssel)	bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
5 SOFTWARE TERVEZÉS Objektum orientált analízis Software tervezési modell dinamikus elemei	- rövid ismételés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
6 Hét: SOFTWARE TERVEZÉS ÉS ADATMODELLJEI Software tervezési modell komponens elemei Software tervezési modell telepítés elemei	- rövid ismételés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
7 Hét:SOFTWARE IMPLEMENTÁLÁS Implementálás előírásai Formai követelmények Fejlesztési prioritások Review Statikus és dinamikus ellenőrzés	- rövid ismételés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
8 Hét: SOFTWARE TESZTELÉS Software tesztelés meghatározása Software tesztelési módszerek Tesztspecifikáció és teszteredmény Hiba és változás követő rendszerek Verifikáció nem azonos a validációval	- rövid ismételés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
9 Hét:SOFTWARE FOLYAMAT Tevékenység és folyamat fogalmak meghatározása Ismertebb folyamatok ismertetése Vizes és társai folyamatok ismertetése Inkrementáló folyamatok ismertetése Iteratív folyamatok ismertetése	- rövid ismételés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
10 Hét: AGILIS SOFTWARE FOLYAMATOK 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	- rövid ismételés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
11 Hét: PROJEKT FOGALMA	- rövid ismételés	

Projekt fogalmának a meghatározása Projekt jellemzői Projekt típusok Projekt életciklusai Projekt folyamatsoportok Projekt tervezés és ütemezés	- a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
12 Hét: KONFIGURÁCIÓ MANAGEMENT Konfiguráció management meghatározása Verziókövetés fogalma Subvesion és ClearCase ismertetése	- rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
13 Hét: MINŐSÉG BIZTOSÍTÁS Minőség biztosítási rendszer meghatározása ISO 9001 rendszer ismertetése Fő és támogató folyamatok CMMI ismertetése	- rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
14 Hét: IT PROJEKT MANAGEMENT Projekt management meghatározása Szervezetek és tervezésük Embortípusok és csapatépítés Motivációs elméletek Rizikó management Konfliktuskezelés Vezetési stílusok	- rövid ismétlés - a hallgatók aktív bevonása az előadásba - történet alapú ismeretátadás - közös megbeszélés - magyarázat-	
Könyvészet 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 Hét: 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 Hét: 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 Hét: 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 Hét: 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 Hét: 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 Hét: Modellező eszköz használata	<ul style="list-style-type: none"> - a hosszú távú 	

	<p>projekthez kapcsolódó elmélet és gyakorlati problémák megbeszélése</p> <ul style="list-style-type: none"> - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése 	
7 Hét: 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 Hét: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 Hét: 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 Hét:UML modell megalkotása, dinamikus 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 	
11 Hét: Esettanulmány: Tesztelés megtervezése	<ul style="list-style-type: none"> - a hosszú távú projekthez 	

	<p>kapcsolódó elmélet és gyakorlati problémák megbeszélése</p> <ul style="list-style-type: none"> - a hallgatók önállóan dolgoznak - az eredmények közös ellenőrzése, megtárgyalása, értelmezése 	
12 Hét Hiba és változás kezelő rendszer 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 	
13 Hét Esettanulmány: Kiértékelés	<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 	
14 Hét: Projekttervező 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 	

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 beépítenek 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
Projekterv készítés	A projekterv megfelel a formai és tartalmi követelményeknek	Írott dokumentum. A dokumentum írásos formai és tartalmi javítása	10%
Software követelményeinek készítése	A software követelmények megfelel a formai és tartalmi követelményeknek	Írott dokumentum. A dokumentum írásos formai és tartalmi javítása	20%
Software architektúrájának készítése	A software architektúrája megfelel a formai és tartalmi követelményeknek	Írott dokumentum. A dokumentum írásos formai és tartalmi javítása	10%
Software design-ának készítése	A software design –ja megfelel a formai és tartalmi követelményeknek	Írott dokumentum. A dokumentum írásos formai és tartalmi javítása	20%
Software tesztspecifikációjának készítése	A software tesztspecifikációja megfelel a formai és tartalmi követelményeknek	Írott dokumentum. A dokumentum írásos formai és tartalmi javítása	10%
Írásbeli vizsga	A vizsgafeladat helyes megoldása	Írásbeli megoldás. Az elméleti tudás gyakorlati alkalmazásának ellenőrzése	30%
10.6. A teljesítmény minimumkövetelményei			
<ul style="list-style-type: none"> • Software fejlesztéshez szükséges tevékenységek, követelmények, tervezés, ellenőrzés gyakorlati megvalósítása megfelelő software eszközzel. 			

Kitöltés dátuma

2013.04.24

Előadás felelőse

Dr. Barabás László

Szeminárium felelőse

Dr. Barabás László

Az intézeti jóváhagyás dátuma

2013. szeptember 30.

Intézetigazgató,

Szenkovits Ferenc