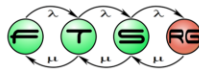


Intelligens rendszerfelügyelet (BME VIMIA370)



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék



Utolsó módosítás: 2013. 02. 11.

Bemutakozás



Dr. Pataricza András



Micskei Zoltán
(IB417, micskeiz AT mit.bme.hu)



Kocsis Imre



Szatmári Zoltán

A tárgygal a kapcsolatos bármilyen kérdéssel keressétek Micskei Zoltánt.

Bemutakozás (oktatók)

- Hibatűrő rendszerek kutatócsoport (FTSRG)
 - kb. 20 kutató-oktató
 - modell alapú tervezés, fejlesztés
 - informatikai rendszerek verifikációja és validációja
 - informatikai infrastruktúrák analízise és szintézise
- Számos nemzetközi és ipari kutatási projekt
- IBM és Intel labor, VMware partner, VCL cloud
- <http://www.inf.mit.bme.hu>



Weboldal: <http://www.inf.mit.bme.hu>

Honlap

<https://www.inf.mit.bme.hu/edu/courses/irf/>
Hírek RSS, előadás anyagok, HF tudnivalók

Hibatűrő Rendszerek Kutatócsoport
Mérés-technika és Információs Rendszerek Tanszék

Közölgés Események Oktatás Kutatás Magunkról

Home » Virtuálizációs technológiák és alkalmazásai »

Intelligens rendszerfelügyelet

Tárgylefelelős: [Micskei Zoltán](#) **Oktatók:** [Kocsis Imre](#) [Pataricza András](#) [Szatmári Zoltán](#) **Korábbi oktatók:** [Tóth Dániel](#)
Tárgy adatai: <https://www.inf.mit.bme.hu/kepesseganyag/VIMA370/>

A tantárgy célkitűzése: A tantárgy célja a hallgatók megismertetése az IT rendszer- és szolgáltatásfelügyeleti módszerekkel. Egy kisvállalati számítógépes infrastruktúra példáján keresztül bemutatjuk az alapvető IT vezéresi és üzemeltetési problémákat, majd megvizsgáljuk, hogy ezeket hogyan lehet jól definiált, automatizálható folyamatokká szervezni. A tantárgy külön hangsúlyt fektet a szoftverfejlesztés és az üzemeltetés kapcsolatára.

A tantárgy tematikája

A félév során a következő témákat és kapcsolódó technológiákat fogjuk megismerni. A témák egyes részének mélyebb elsajátítását gyakorlati házi feladatok segítik.

Tématerület	Témák	Módszerek, technológiák
Bevezető	Bevezető és tantárgy ismertető Informatikai rendszerek alapelemei	
Modellezés	Modellezési célok és módszerek Adatmodellek készítése Üzemi és üzemeltetési folyamatok	UML2, metamodellezés
Szriptelés	Szript nyelvek, Bash és PowerShell alapjai	Bash, PowerShell

Weboldal: <https://www.inf.mit.bme.hu/edu/courses/irf/>

Honlap: bejelentkezés

BME Címtár belépés

Budapesti Műszaki és Gazdaságtudományi Egyetem
Címtár

Belépés

A belépéshez a címtáras azonosító és jelszó megadása szükséges. Ha Ön szerepel a Neptunban, és még nem állított be címtáras jelszót, kérjük, tegye meg [ezen az oldalon](#).

Sikeres belépés után az azonosítást kérő oldalra irányítjuk vissza. Az azonosítást a következő oldal kérte: [https://www.inf.mit.bme.hu /shibboleth](https://www.inf.mit.bme.hu/shibboleth)

Felhasználónév: @bme.hu

Jelszó:

© BME

A honlaphoz való bejelentkezéshez a központi BME Címtár azonosítót lehet használni.

Bővebben lásd: <https://www.inf.mit.bme.hu/wiki/it/szolgáltatások>

IRF Q&A oldal (beta)

- Levlista helyett
- <http://q2a.inf.mit.bme.hu/>

The screenshot displays the IRF Q&A website interface. At the top, there is a navigation bar with links for 'Questions', 'Unanswered', 'Tags', 'Users', 'Admin', and 'Badges'. A search bar is located on the right side of the navigation bar. Below the navigation bar, the main content area is divided into two columns. The left column is titled 'Recent questions and answers' and lists four questions with their respective scores and answers. The right column is titled 'Most popular tags' and lists 'winrm' and 'powershell'. Below the tags, there is a section for 'All categories' with 'IRF tantárgy (2)' and 'Q&A web (2)'. At the bottom of the page, there is a footer with the text 'Help get things started by asking a question.' and a small logo for 'MÉGEVÉSEM 1782'.

- Q&A oldalak: <http://stackoverflow.com/about>

Webcím: <http://q2a.inf.mit.bme.hu/>

A levelező listákon könnyen elveszik egy-egy gyakori probléma megoldása, ezért most kipróbálunk egy Q&A oldalt, amiben a tárggyal kapcsolatos kérdések és azok helyes válaszai sokkal könnyebben megtalálhatóak. Aki nem használt még ilyen oldalt, annak érdemes megnézni a Stack Overflow leírását.

A félév menete

▪ Előadások:

- Minden héten: hétfő (10-12)
- Páratlan héten: szerda (8-10)

▪ Gyakorlatok:

- Páros héten: csüt. (8-10) vagy péntek (10-12)
- **Opcionális**, gyakorlatok anyaga otthon is elvégezhető

▪ Fogadóóra:

- Micskei Zoltán: csütörtök 16:15-17:00

▪ Labor: Informatikai technológiák labor I.

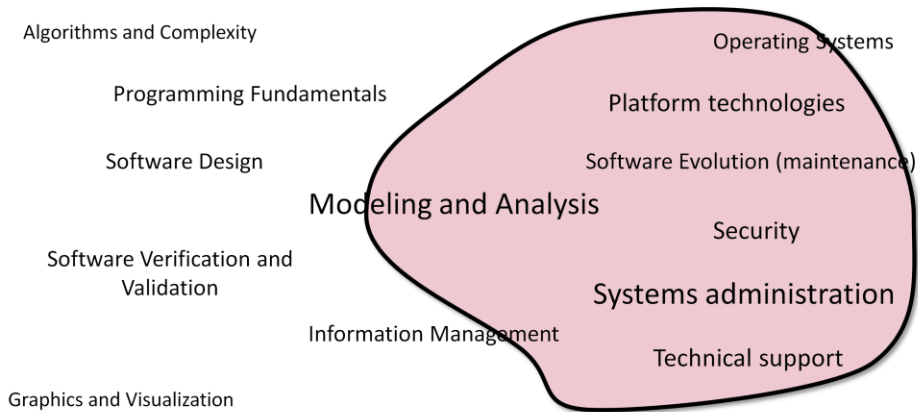
- AUT + IIT + MIT mérések, beosztás a weben lesz majd

Tartalom

- Bemutatkozás
- **Tematika**
- Szeminárium
- Követelmények

Az informatika területei









ACM & IEEE Computing Curricula (részlet)



ACM Computing Curricula: <http://www.acm.org/education/curricula-recommendations>

Az informatika rendkívül széles spektrumú, ebből az eddigi tárgyak jó néhány területet alig érintettek még. A tárgy a megjelölt területekkel foglalkozik majd.

Tematika

	Infrastruktúra alapelemek (1E)	
	Modellezés (1E+1GY)	V
	Szkriptelés alapok (3E+1GY)	
	Címtárak (3E+1GY)	HF1
	Konfigurációkezelés (3E+1GY)	HF2
	Monitorozás (3E+1GY)	HF3
	Szolgáltatásbiztonság, hibatűrés (2E+1GY)	V
	Virtualizáció és cloud computing (3E)	V

A félév folyamán ezeket a területek fogjuk érinteni.

V – kapcsolódó gyakorlati feladat a vizsgán

Előkövetelmények (témák)

- **Operációs rendszerek**
 - OS felépítése, szerepe
 - Felhasználókezelés, biztonság
- **Számítógép hálózatok**
 - TCP/IP alapok
 - DNS
- **Szoftvertechnológia és Szoftvertechnikák**
 - Szoftver modellezése, UML
 - Szoftver architektúrák
- **Programozás tárgyak (Java, C#)**
- **Mérés 4. és IRÜ: kapcsolódik, de sajnos párhuzamos**



Ezekben a tárgyakban tanultakra építünk, főleg a fent megjelölt területek lesznek azok, amiket érdemes feleleveníteni.

Ki mit tanulhat itt?



Szoftverfejlesztő

Ki mit tanulhat itt?



Szoftverfejlesztő

- Felügyeletre tervezés
- Nagyvállalati IT elemei
- Szoftverfejlesztési ciklus maradék elemei

Jó szoftvert csak úgy lehet írni, ha ismeri a fejlesztő, hogy később az az alkalmazás milyen környezetben fog működni. Már a legelső rendszerterveknél érdemes bevonni a leendő üzemeltetőket, és elgondolkodni, hogy hogyan lehet majd egyszerűen üzemeltetni az adott alkalmazást, beilleszteni az IT környezetbe. Ehhez viszont fontos, hogy egy nyelvet beszéljenek a fejlesztést és az üzemeltetést végzők, ne teljesen szeparált folyamatokban gondolkodjanak.

Példa: MS Common Engineering Criteria

- Követelmények a szervertermékekkel szemben:
 - Állapotmodell definiálása szabványos modellel
 - Állapotok, események, teljesítményszámlálók...
 - Management Pack
 - Illesztés a rendszerfelügyeleti eszközökhöz
 - Virtualizáció támogatása
 -
- Ezeket már a tervezésnél figyelembe kell venni!



Forrás: Microsoft Common Engineering Criteria,
<http://www.microsoft.com/cec/en/us/cec-overview.aspx>

Egy csomó olyan követelmény van a fejlesztőkkel szemben is, amihez tudni kell, hogy hogyan fogják majd ezeket a rendszereket üzemeltetni. Ismerni kell, hogy mire jó egy központi címtár, hogyan lehet csatlakozni pl. egy Active Directory-hoz, milyen következményei vannak annak, ha virtuális gépben fut a szerver, amin az alkalmazás később lesz, hogy néz ki egy monitorozó alkalmazás, ami később felügyelni fogja az alkalmazásunkat.

Ki mit tanulhat itt?

- Üzemeltetési körkép
- Hasznos eszközök, alaptechnológiák
- Rendszermenedzsmet újabb irányzatai

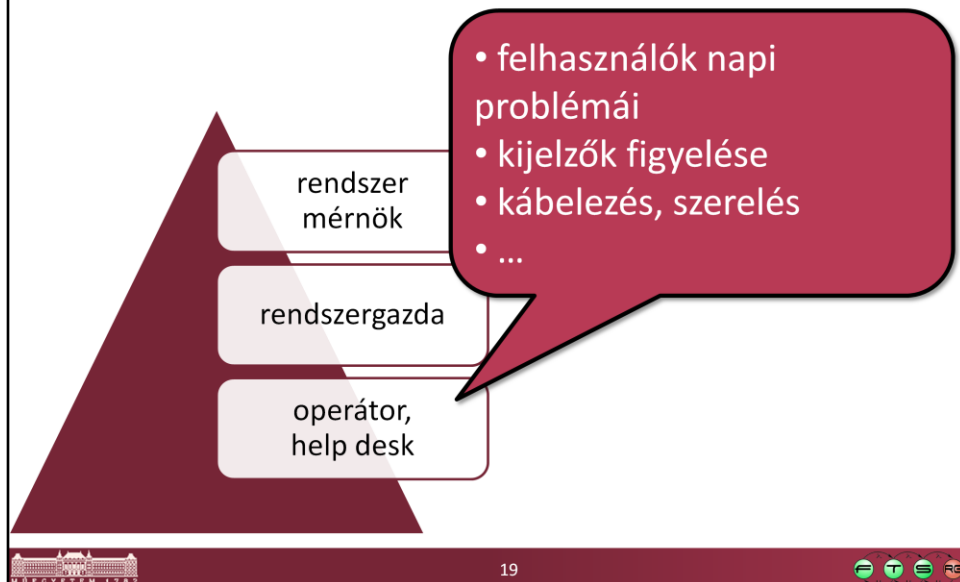


Rendszermérnök

Messze nem fogunk minden üzemeltetési területet érinteni a félév folyamán, csak párat kiemelünk.

A hangsúly azon lesz, hogy az általános probléma és az alap technológiák után elgondolkodjunk kicsit, hogy mi is az adott terület általános folyamata, hogyan lehet azt automatizálni, „intelligenssé” tenni.

A rendszerüzemeltetés szereplői



A felosztást és a pontos elnevezést mindenki máshogy használja, de mindenhol megvannak a szintek. A fólia mondanivalója annyi, hogy az üzemeltetésen belül is vannak különböző szintek, messze nem csak rendszergazdaságról szól ez a szakma.

A rendszerüzemeltetés szereplői

rendszer
mérnök

rendszergazda

operátor,
help desk

- szerver karbantartás
- teljesítményfigyelés
- új szolgáltatás telepítése
- ...

A rendszerüzemeltetés szereplői

rendszer
mérnök

rendszergazda

operátor,
help desk

- folyamatok megtervezése
- működés finomhangolása
- speciális problémák
- ...

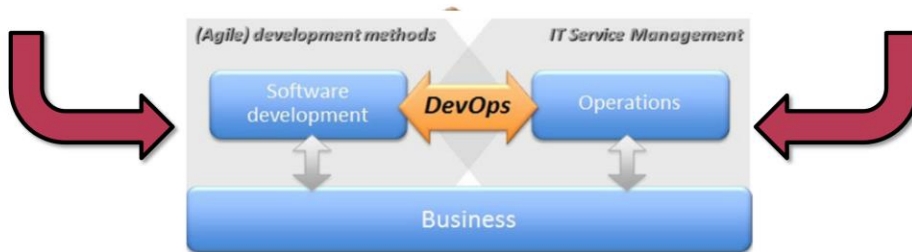
Ki mit tanulhat itt?



Szoftverfejlesztő



Rendszermérnök



Képek forrása:

- Damon Edwards. What is DevOps all about? <http://dev2ops.org/2010/02/what-is-devops/>
- Michael Hüttermann. DevOps for Developers, Apress, 2012.

Tényleg kell ez?

Álláshirdetések:

Skills Desired

- Experience with at least one OO language eg. Java, C, C++ or C#
 - Database knowledge (SQL)
 - Basic operating system knowledge (Windows and Unix) / user level experience
- Script language knowledge is an advantage (e.g. Perl, Shell)

Morgan Stanley

Preferred Qualifications:

Google

- MS/PhD in Computer Science or equivalent.
- Experience in development and/or test automation.
- Excellent coding skill in C, C++, Java or Python.
- Scripting skills in Python, Perl, Shell or another common language.
- Extensive knowledge of UNIX/Linux environments.
- Deep knowledge of internet technologies.

Senior infrastructure (devops) engineer

Apply Now

Coders | Budapest

Send Jobvite

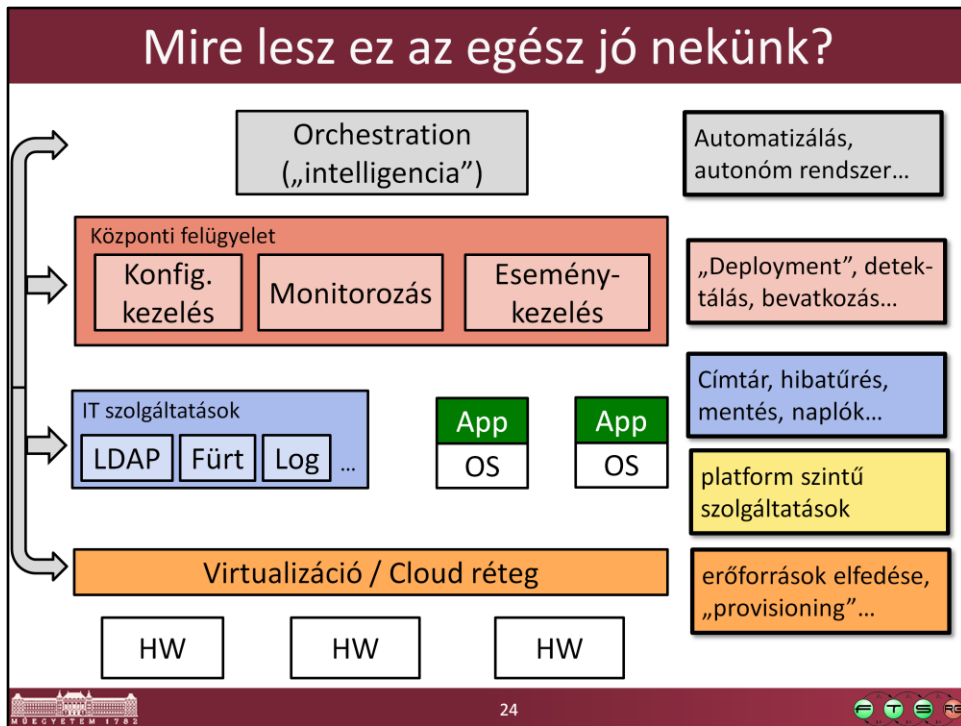
We're looking for passionate and skilled infrastructure (devops) engineers to help us build and scale the backend system behind prezi.com and the infrastructure of our continuous integration pipeline. This position requires a backend developer with strong sysadmin skills or a sysadmin with a considerable web (backend) developer experience.

Prezi



Google, Morgan Stanley, Prezi álláshirdetések

Érdeemes körbenézni a nagyobb cégeknél, hogy milyen ismeretek jelentenek ott pluszt.



Ha nem egy játék alkalmazást készítünk, hanem egy nagyvállalati környezetben működő rendszert / kritikus rendszert / igazán nagy méretű webes megoldást.

Az általunk készített alkalmazásoknak együtt kell tudni működni az összes többi komponenssel és alapszolgáltatással ahhoz, hogy tényleg egy jó rendszert kapjunk.

Erre az ábrára még a félév végén visszatérünk.

Tartalom

- Bemutatkozás
- Tematika
- **Szeminárium**
- Követelmények

Szeminárium?

- A gyakorlatok alternatívája
- Előadáshoz kapcsolódó „elméleti” témák
- Maximum 15 fő

- Hangsúly:
 - **Metodológia**
 - **Algoritmika**
 - (Technológia)

Témák

- Mérnöki problémamegoldás és **kényszerkielégítési** problémák
- **Adatbiztonsági követelményrendszerek** és vizsgálatuk
- Felügyeleti mérések **vizuális adatfelderítése**
- Az **IT teljesítményelemzés** alapjai
- Szemelvények a **diagnosztika**-elmélet területéről

Tudnivalók

- Időpont: péntek 10-12 (páros hét)
- Félév teljesítése:
 - Szemináriumi jelenlét
 - HF-ek helyett: 1 db „tanulmány” („term paper”) oktatóval egyeztetett témában (értékelés → HF-pontszám)
- Jelentkezés: email - ikocsis@mit.bme.hu
 - Motiváció
 - Szakiránybekerülési pontszám
 - (Név, Neptun-kód)

Tartalom

- Bemutatkozás
- Tematika
- Szeminárium
- **Követelmények**

Követelmények (kivonat) – aláírás

3 darab kis házi feladat:

- HF-enként: 0-15 pont (idén nincs dokumentáció!)
 - min 40% szükséges minden HF-ből
- eredménye beleszámít a jegybe (45%)
- ~ 1-1,5 hetes beadási határidők
- feltöltés a határidő után automatikusan záródik
- HF1: ~6. hét / HF2: ~8. hét / HF3: ~10. hét
- Linux **ÉS** Windows is
- Szöveges értékelés (tanuljunk az értékelésből!)



30



- A határidő szigorú, a határidő lejártá után már nem lehet leadni házi feladatot (akkor se, ha 1 másodpercet késtél, akkor se, ha összeomlott az otthoni gép, stb.). Hagyjunk magunknak elég tartalékot!
- Miért kell Windowst és Linuxot is használni: ez alapján tudjuk megítélni az egyes platformok képességeit, lesz összehasonlítási alapunk. Pont ez az egyetemi képzés lényege, hogy nem csak egy konkrét megoldást ismerünk meg, hanem általánosabb módszereket.

A pontos követelményeket lásd:

<https://www.inf.mit.bme.hu/edu/courses/irf/kovetelmenyek>

Követelmények (kivonat) – vizsga

- Vizsga: írásbeli + opcionális szóbeli
 - Elméleti kiskérdések, 2 gyakorlati feladat
 - Megtekintés, nem reklamálás
 - Pontvadászat helyett szóbeli javítás (+1/-1 jegy)



Opionális szóbeli: ha valaki úgy érzi, hogy jobbra tudja az anyagot, mint a végén megállapított jegy, akkor a végeleáthatatlan reklamálás és pontvadászat helyett inkább jöjjön el a megtekintésre szóban javítani.

Követelmények (kivonat) – pótlás

- Nem leadott vagy elégtelen HF-eket pótolni kell
- Mindegyik HF pótolható
- Pótlás különjárási díj köteles (1800 Ft)
- DE: Pótlásnál **új feladatot kell megoldani**
 - Különben nem fair azokkal, akik időben leadták
- Elfogadott HF-et lehet javítani
(ezt is a pótlási héten)

Házi feladatok céljai

- 3. éves mérnökhallgatóknak szól a tantárgy
- Tapasztalat és ipari visszajelzés:
 - Bizonyos képességek hiányoznak a végzősökből

Önálló problémamegoldás

- Előadás: kérdések és irányok bemutatása
- Gyakorlat: technológia alapok kipróbálása
- HF: egy kis szelet **önálló megismerése**

HF értékelési szempont: igényes kód

Értékelés: igényes kód

Interfészek betartása, jó paraméterkezelés, ellenőrzött bemenet, konstansok, hatékony lekérdezések, hibakezelés, tesztesek, kommentezés, helyes tagolás... -> lásd a [weboldalt](#) (10+8 tanács)

Ellenőrző lista

A következő ellenőrző listán minden leadás előtt érdemes végigvenni.

MUST (kötelező, súlyos hibát jelent)	
SZ1 Szintaktikai hibás megoldást nem értékelünk.	<p>A Java/ NET kódok esetén a javítás során a leadott kódot mindig teljesen újrafordítják a javítók. Így leadás előtt érdemes ellenőrizni azt, hogy tényleg minden fájlt beadunk-e, ami a fordításhoz kell, és esetleg más könyvtárba rakva/más gépen fordítva is lefordul a kód.</p> <p>Ugyanígy szkript esetén, ha a feladatírásban megadott példa paraméterezés esetén a parancsértelmező szintaktikai hibára panaszkodik (ismeretlen kulcsszó, hiányzó zárójel stb.), akkor általában nem is nézzük tovább a megoldást.</p> <p>Két tipikus hiba szokott lenni ezzel kapcsolatban. Az egyik, hogy a saját környezetben tényleg lefordult, azonban nem dokumentáltuk pontosan, hogy milyen verziójú fordítót vagy parancsértelmezőt használtunk (nem elég az, hogy PowerShell kell a szkripthez, hanem adjuk meg a pontos verziót is, pl. PowerShell 3.0).</p> <p>A másik tipikus hiba, hogy a leadás előtt egy perccel még átfomázzuk kicsit a kódot, pár sortörést rakunk bele, majd nem futtatjuk le még egyszer a szkriptet az összes teszttel. Ne tegyük! Ne adjunk le olyan módosított kódot, amit nem fordítottunk és teszteltünk le újra!</p>
SZ2 A programnak pontosan a specifikált interfészt kell nyújtania (elnevezés, bemeneti paraméterek neve és száma stb.)	<p>A szkriptnek, programnak betűre pontosan azt a fájlnevet és paraméterelnevezést kell használnia, ami a kiírásban van. Hasonlóan a be- és kimeneti fájloknak is a megadott formájúnak kell lennie.</p> <p>A külső interfész pontos megvalósítása az egyik legkritikusabb pont egy programnál, ez alapján tudják majd mások használni a programunkat. Ha nem a specifikált elnevezést vagy paraméterezést használjuk, akkor nem fogják tudni használni a programunkat. Ráadásul a házi feladatok javítása részben automatizáltan történik, így például ha elfut a szkript nevével vagy valamelyik paraméterét, akkor a javítónak kézzel kell majd nekélteni futtatni a megoldásunkat, amit nagyon nem szeretünk.</p> <p>Többször is ellenőrizzük, hogy pontosan a kiadott specifikációt valósítjuk-e meg, és a feladatírásban megadott módon futtatható-e a megoldásunk!</p>
SZ3 A kódnak nem csak a	<p>A kódot mindig ellenőrizzük úgy is, hogy egy friss parancssorból próbáljuk meg kézzel elindítani.</p>



A részletes házi feladat tanácsokat és irányelveket lásd:
<https://www.inf.mit.bme.hu/edu/courses/irf/hazifeladat>

Mit kéne egy szoftverfejlesztőnek tudni?

- Hasznos lista: [Programmer Competency Matrix](http://www.indiangeeek.net/programmer-competency-matrix/)
 - Algorithms, code organization, IDE, requirements, database, platform internals...

build automation	Only knows how to build from IDE	Knows how to build the system from the command line	Can setup a script to build the basic system	Can setup a script to build the system and also documentation, installers, generate release notes and tag the code in source control
automated testing	Thinks that all testing is the job of the tester	Has written automated unit tests and comes up with good unit test cases for the code that is being written	Has written code in TDD manner	Understands and is able to setup automated functional, load/performance and UI tests
code readability	Mono-syllable names	Good names for files, variables classes, methods etc.	No long functions, comments explaining unusual code, bug fixes, code assumptions	Code assumptions are verified using asserts, code flows naturally – no deep nesting of conditionals or methods
defensive coding	Doesn't understand the concept	Checks all arguments and asserts critical assumptions in code	Makes sure to check return values and check for exceptions around code that can fail.	Has his own library to help with defensive coding, writes unit tests that simulate faults



Programmer Competency Matrix, <http://www.indiangeeek.net/programmer-competency-matrix/>

Érdemes végignézni a listát, hogy mindenki be tudja lőni a saját tudásszintjét, és lássa, hogy mit lehetne még fejleszteni.

(Forrás: <http://balassygyorgy.wordpress.com/2012/06/21/allasinterju-fejleszttoi-kompetenciak/>)

(Dokumentáció – érdemes tudni)

- Része minden mérnöki munkának
- Olyan információ, ami nincs benne a kódban
- Ez „adja el” a munkánkat
- Tanulni, gyakorolni kell
 - Tévhit: „Majd élesben tudok jót is írni...”

Formai, tartalmi tanácsok:

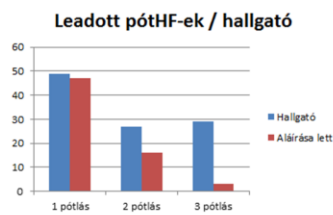
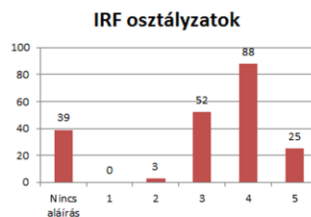
<http://www.inf.mit.bme.hu/edu/other/documentation>



Idén nem kérünk ilyen dokumentációt a házi feladathoz, de azért érdemes megnézni a fenti linket, önálló laborhoz vagy a későbbiekben még jól jöhet.

Nehezek ezek a követelmények?

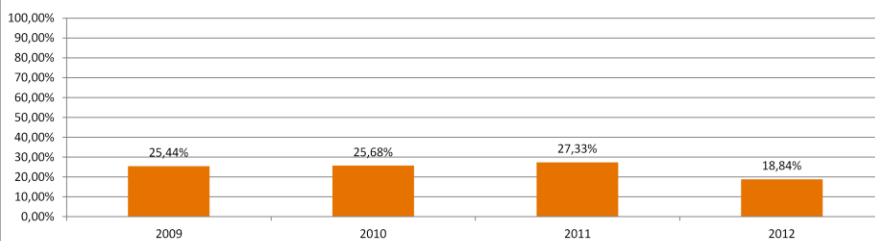
- Nem teljesíthetetlen, csak más, mint eddig
- HF a szűrő, problémák:
 - másolás
 - igénytelen kód
 - specifikációtól eltérés
- Tapasztalat:
 - 2-3 HF már nem pótolható
- **Figyelem:**
 - Nincs keresztfélév
 - Előkövetelmény a szakdolgozathoz az IRF jegy



Figyelem: az előtanulmányi rendtől még méltányossággal se lehet eltérni, így ha valaki nem teljesíti az IRF-et, az leghamarabb az aktuális utáni 3. félévben veheti fel a Szakdolgozat tantárgyat!

Nehezek ezek a követelmények? (számok)

(Nem teljesítette) / (Összes hallgató)



2009

- 4 darab HF
- Két alfeladat mindenhol

2010

- 4 darab HF
- Két alfeladat mindenhol
- Kiadott VM
- Tavalyi tapasztalat

2011

- 3 darab HF
- Egy alfeladat mindenhol
- ~100 oldal segédlet

2012

- 3 darab HF
- Egy alfeladat mindenhol
- ~100 oldal segédlet
- Opcionális gyakorlatok



Nehezek ezek a követelmények? (hallgatók)

„...úgy éreztem, hogy ez a tárgy az egyetem egyik legnehezebb tárgya”

„A házikkal sok munka volt, több mint egy átlagos aláírás szerzésért szokott lenni. ”

„Olyan dolgokat tanultunk, melyekhez - véleményem szerint - mindenkinek érdemes értenie, aki informatikus.”

„A házi feladatokkal ment el sok idő, de megérte.”

„A tárgyat a megfelelő energia befektetéssel könnyedén el lehetett végezni. Tényleg csak annyira volt szükség, ami elhangzott előadáson is - időben neki kell feküdni a háziknak.”

„Nagyon tetszett ez a szemlélet amit a tárgy keretein belül megismerhettünk.”



Szemeztetések OHV visszajelzésekből

Nehezek ezek a követelmények? (oktatók)

- Igen...
- ~560 kijavítandó HF (6 javítóóra)
 - Megnézni és kipróbálni egy szkriptet / programot
 - Szöveges visszajelzés mindegyikre
- ~670 munkaóra = 3,9 mérnökhónap (!)
- viszont ebből tanulhattok a legtöbbet...



Az elkészült munka stílusára, hatékonyságára, érthetőségére, minőségére, pontosságára is kaptok visszajelzést, és nem csak egy jegyet vagy egy pontszámot. Érdemes ezeket higgadtan, alaposan elolvasni, és tanulni belőlük.