

# Üdvözli Önöket A Programozás Technológiai Eszközei c. tantárgy!!

a.k.a: Szoftverfejlesztési projektmunka a gyakorlatban II. (PGY2)

Bakay Árpád dr.  
NETvisor kft  
(30) 385 1711  
[arpad.bakay@netvisor.hu](mailto:arpad.bakay@netvisor.hu)

# Menetrend

- Általános tudnivalók
  - Miről szól ez a tantárgy?
  - Órarend, követelmények, vizsga, konzultációk
- Mai anyag:
  - Mik a „programozás technológiai eszközei”?
  - A szoftver
  - A projekt
  - Projekt tervezés és ütemezés
  - A szoftverfejlesztés mint hivatás

# Magamról és a NETvisorról

## Magamról

- 1984 - 88 - 91 BME Villamosmérnöki kar
- Szoftverfejlesztés, rendszerintegráció, kutatás
- Oktatás, óraadóként
  - 1998-tól: BME, Vanderbilt (USA), NYME (Sopron),
  - 2005-től : ELTE

## NETvisor

- „Informatikai és Kommunikációs Szolgáltató Kft.”
- Távközlési és nagyvállalati informatika
- Kutatási és R&D projektek egyetemekkel

# Miről lesz szó?

- Egyrészt: A professzionális szoftverfejlesztést támogató szoftver eszközökről
  - A módszertanok már ismertek az Önök számára
  - Kurrens, korszerű eszközöket tanulunk
  - Cél az eszközök gyakorlati használatának elsajátítása
- Másrészt: Korszerű szoftver környezetekről
  - Java EE, alkalmazáserverek, architektúrák
  - Végig egyetlen viszonylag komplex alkalmazással foglalkozunk, ezen dolgozgatunk
- *Csak ami a gyakorlatban is hasznos!!!*

# Általános információ

- Ld. „syllabus”
- Website

<http://telco.ikk.inf.elte.hu/twiki/bin/view/Main/WebHomePGY2>

- Ami a website-on megjelenik, arról Önöknek további értesítés nélkül tudni kell!
  - A kedden számonkért anyag legkésőbb előző pénteken felkerül!
- Órák időpontja
  - Kedd 12:00:00-13:30:00
  - Összesen 12 alkalom
  - (Április 3-én és 10-én nincs óra)
  - Vizsgaidőszak: Május 15-én vizsga ZH 4 órában!!

# Általános információ – folyt.

- Könyvek e témában jórészt elavultak!
  - ...és drágák
- Könyvek helyett:
  - Java tutorialok: <http://java.sun.com>
  - Link gyűjtemény a PGY2 website-on
  - Egyéb internetes anyagok

# Számonkérés

- Kis ZH-k: rendszeresen!!
  - Kb. 2 hetente
  - 168 órán belül pótolható
- Házi feladat: 2 részletben
- Nagy vizsga-ZH
- „Honor code”, szankciók

# Első rész: Alapfogalmak

- Szoftverfejlesztés és technológia
- Szoftvertechnológiai feladatok és eszközök
- Futtatókörnyezetek szolgáltatások

# 1. A szoftverfejlesztés technológiája

*Itt az önvizsgálat ideje!*

## A szoftverek

- Megfelelnek-e a követelményeknek?
- Menteselek-e a hibáktól? Ki lettek-e jól tesztelve?
- Áttekinthető-e a belső struktúrájuk?
- Karbantarthatók-e?
- Felhasználhatók-e más projekteken?
- Időben elkészültek-e?

# Programozás és technológia

- Technológia: gyártási folyamat elmélete és gyakorlata
  - Szisztematikus, szabályozott, ellenőrzött
  - A legmegfelelőbb eszközökkel
  - Üzemszerű „termelés”
  - Költség-, minőség-, időráfordítás kézbentartása, optimalása
- Miért kell mindez a szoftverfejlesztéshez?
  - A bonyolultság miatt az „ad-hoc”, ösztönös megközelítés hamar eléri lehetőségek határát
  - A hatékony csapatmunka, a minőség és a hosszabb távú karbantarthatóság fegyelmet, további odafigyelést igényel.
- Angol neve: **Software Engineering**
- *Magyarországon ez nem az általános megközelítés*
  - *„Nemzeti sajátosság”, amely érezhető hátrányt jelent*
    - *Előttünk járnak: Csehország, volt Szovjetúnió, India*

# Milyen a jó szoftver?

- Hibamentes, megbízható
- Hatékony
  - Skálázható: nagyobb terheléshez (több feladat, funkció/perc) csak arányosan nagyobb hardver kell
- Jól használható
  - Könnyen kezelhető
  - Toleráns a felhasználói hibákra
- Áttekinthető, moduláris, ettől
  - Karbantartható,
  - Újrafelhasználható
- Szabványokhoz illeszkedő
  - Más szoftverekkel képes együttműködni
- ...és megint csak: **időben elkészül!**

# A szoftver költsége

- A legtöbb rendszerben a hardvernél is jóval drágább
  - Kivéve: tömegtermékek, ahol a sw. fejlesztés költsége szétoszlik
- Fejlesztés csak kb. 40% - a többi karbantartás

# A fejlesztés üzleti modelljei

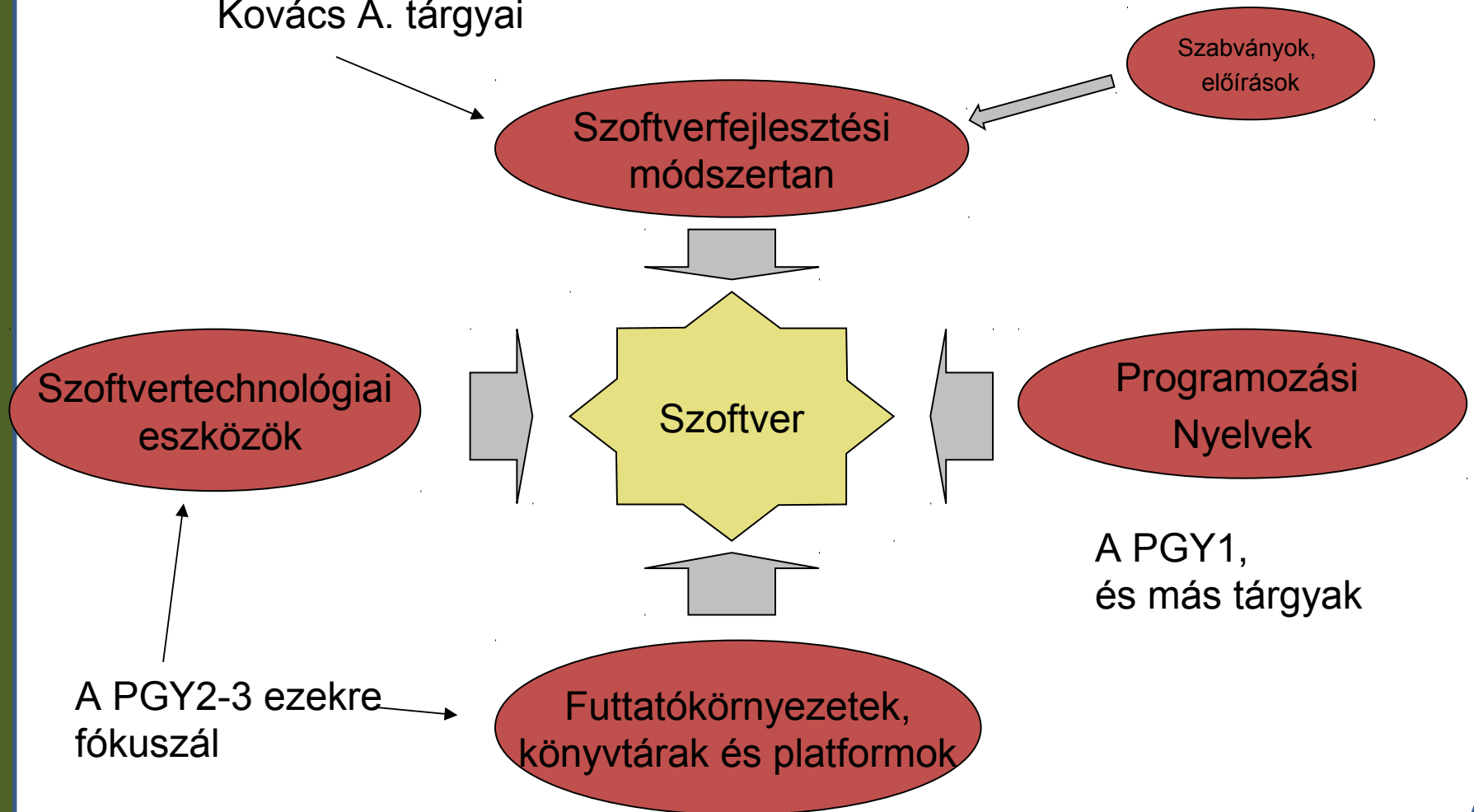
- Egyedi megrendelésre vagy általános használatra/értékesítésre
- Mindkét esetben a szoftverhez értendő a program kód és bináris mellett
  - a dokumentáció
    - felhasználói, üzemeltetői/adminisztrátori , fejlesztői
  - a tesztek
    - teszt jegyzőkönyvek, automatikus teszt programok

# Szoftverfejlesztés emberi tényezői

- Többféle résztvevő együttműködése: vezetők, tervezők, fejlesztők, tesztelők
  - Vezetői feladatok: időzítések, munka szétosztása, költségek figyelése
  - A tervezői és fejlesztői feladatok kevésbé szétválaszthatók, mint más jellegű projekteknél
- A munka erősen szellemi jellege miatt fokozottan szükséges a sikerhez:
  - bizalom, lelkesedés, sikerélmény, személyes igények kielégítése (pénz, környezet, kikapcsolódás, tanulás, stb.)

# A szoftverfejlesztés technikai tényezői

Langer T.,  
Kovács A. tárgyai



## 2. Szoftvertechnológiai eszközök

*Szoftverek, amelyek támogatják a tervezési és fejlesztési módszertanok követését, ill. javítják a végtermék (szoftver, dokumentáció és tesztek) jó minőségét, karbantarthatóságát.*

(Általában NEM tartoznak ide az alapvető fejlesztőeszközök: compiler/linker, debugger, kód editor stb.)

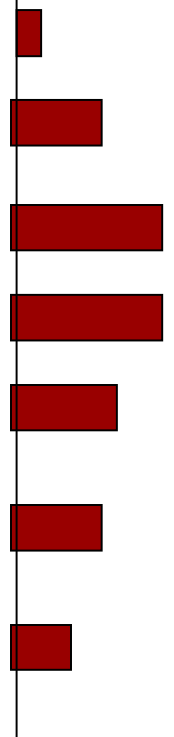
### Computer Aided Software Engineering

A szónak van egy szűkebb értelmezése is, ti. leggyakrabban a korábban nagyon ígéretes UML-szerű fejlesztőeszközökre használták („CASE eszközök”).

# Szoftverfejlesztés jellemző feladatai

- Projekt tervezés és irányítás
- Követelmény-elemzés
- Tervezés, modellezés
- Kód előállítás és hibajavítás
- Tesztelés
- Konfiguráció-, változás- és feladatkezelés
- Dokumentáció generálása

Hangsúly  
nálunk



*Mindegyik feladattípushoz megismerünk majd eszközöket.*

# Piaci körkép (korántsem teljes!!)

	Drága		Olcsóbb		Ingyenes!	
	Rational IBM	Telelog ic	Borland	Microsoft	Open Source/ generic	Open Source/ Java
Projekt Managemen t		Dash Board	Tempo	MS Project	Planner, DotProject	OpenProj Gantt
Követelmény- kezelés	Requisite Pro	Doors	Caliber RM	SharePoint/ Blackbot	Use Case Maker	
Modellezés	Rose, Software Architect	Tau	Together	Visual Studio for Architects	Umbrello	Eclipse MF, Fujaba, AndroMDA
Konfig. manageme nt	ClearCase	Synergy	StarTea m	Visual SourceSaf e	CVS, Subversion	JCvs, JSvn
Feladat- és hibakezelés	ClearQue st	Synergy	StarTea m	Team Foundation Server	Bugzilla, Bugtracker.NE T	Scarab, Jtrac, BugTracker
Tesztelés	Purify, TestSuite	Logi scope	SilkTest	VisualStudio Test Framework	xUnit	Junit, HttpUnit
Dokumentáció	SoDa	Doc Express	DocFlex	SandCastle	DoxyGen	javadoc

Egy hasznos link: <http://java-source.net/>

# 3. Futtatókörnyezetek

Szoftverek (környezetek, framework-ok, platformok, könyvtárak) amelyek szolgáltatásokat nyújtanak a felettük futó alkalmazások számára.

A futtatókörnyezet maga is futhat egy másik környezetben. Az op.rsz, sőt a hardver is egyféle „futtatókörnyezet”.

Modern környezeteknél az alkalmazás felől a kötődés egyre erősebb, erre mondják, hogy az alkalmazás nem a környezet „felett”, hanem „benne” fut.



# Általános célú futtatókörnyezetek - jellemző szolgáltatási szintek

- Minimálszolgáltatások (CP/M, MS-DOS)
  - Processz indítása és leállítása
  - User konzol interfész
  - File I/O
  - Simple memory management
  - Időkezelés, időzítés
- Normál, „XX. szd.-i” (Windows+COM, Unix+X, **Java SE**)
  - Hálózati kommunikáció
  - Multitasking, IPC (szemaforok, shared memory, stb.)
  - Virtual memory management
  - Grafikus UI, ablakok
  - Multithreading
  - Remote procedure/method call (RPC, Java RMI)

# (folyt)

- „Premium” (Corba, **Java EE**, Spring, MS .NET, Cake PHP/Zend, Ruby on Rails)
  - tranzakció-kezelés,
  - persistence, caching
  - security
    - Encryption
    - AAA (authenticate, authorize, audit)
    - single sign-on
  - load-balancing, failover,
  - messaging
  - configuration, diagnostics, management
  - Logging
  - Belső modulok/komponensek életciklusának és kapcsolatainak vezérlése
    - „Komponens framework”

# Az futtatókönyezetek típusai

- Backend (GUI nélküli) frameworkok
  - Java EJB, Spring, Corba, DCOM
- Hagyományos GUI alkalmazás frameworkok
  - MFC, Swing, Eclipse RCP
- Web-alkalmazás frameworkok
  - Pl. Zope, Cake PHP, JSF, Tapestry
- Browser-oldali alkalmazás frameworkok
  - Dojo, ExtJS, JQuery, Google Web Toolkit

# Java Enterprise Edition

*Erről fogunk elsősorban tanulni*

- A Java világ 'prémium' futtatókörnyezete
  - Mindent nyújt a prémium kategóriában
  - Plusz:
    - környezet Web alkalmazások számára
    - Szabványos, eszközfüggetlen adatbázis kapcsolat
  - Maga a Java Standard Edition környezetben fut.
  - További, Java EE-re épülő környezetek: JSF, Spring, Oracle ADF

# Második rész: A szoftverfejlesztési projekt

- Projekt, azaz:
  - Egyetlen, egyedi eredményre irányul
  - Több különböző tevékenységet végző résztvevő közreműködésével
  - Nem folyamatosan, hanem határidőre befejeződik

Ha valami nem projekt, akkor az **folyamat (process)**, pl. cement v. autógyártás

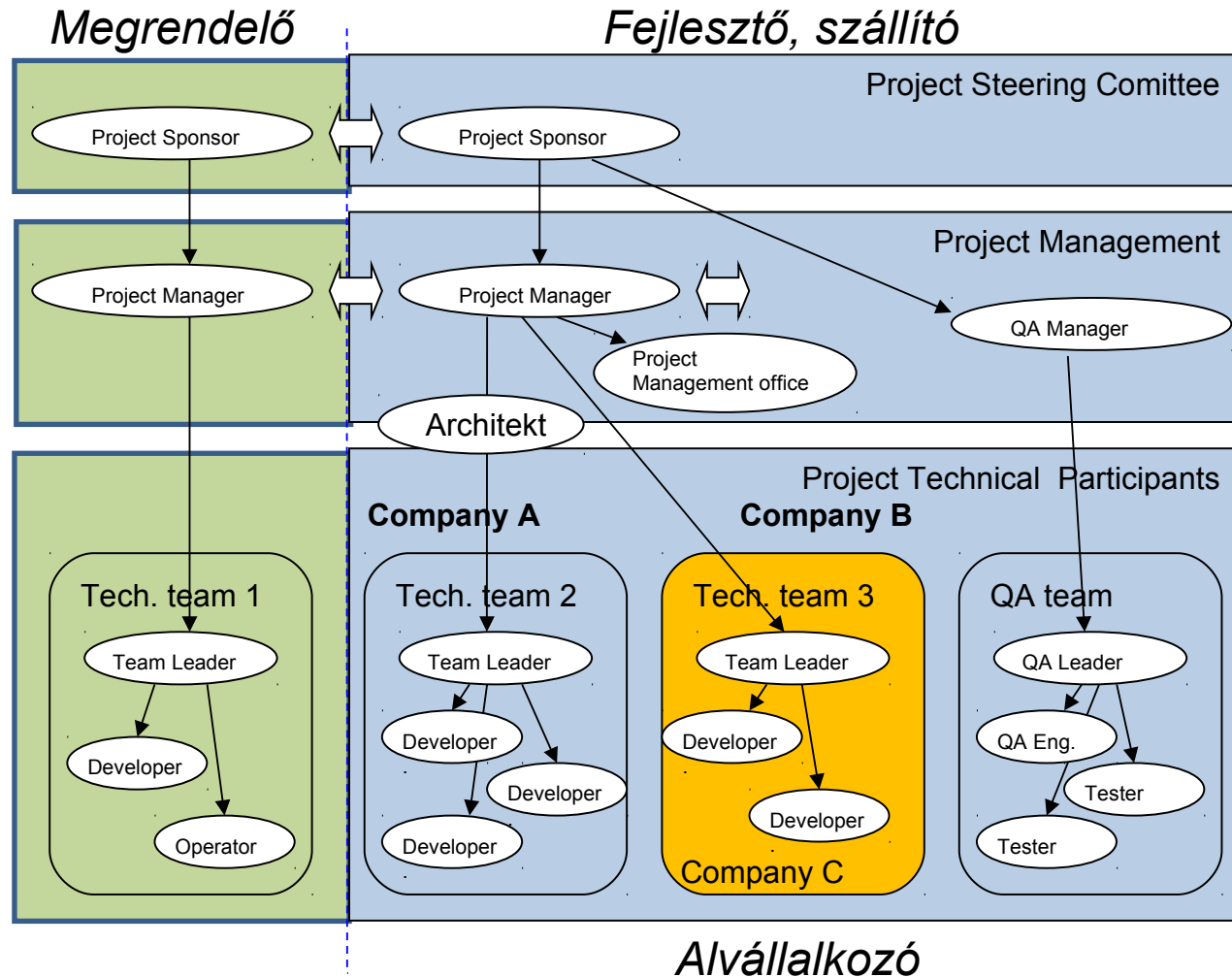
# Ha projekt, akkor...

- Projekt hierarchia: a résztvevő emberek szerepe és alá-fölérendeltsége
- Projekt terv: a megvalósítás egy előre megtervezett munkafolyamat ill. költségvetés szerint zajlik.
- Dokumentumok: a lényeges eredmények, események, előírások dokumentumokban kerülnek rögzítésre. A dokumentumok lehetnek szabad szöveges jegyzetek, rögzített formátumú szövegek, diagramok, táblázatok, modellek stb.
- Átadás: a szoftver (egy vagy több lépésben) végül átadásra kerül egy projekten kívüli fogadónak, és ezzel a projekt, betöltve hivatását, szükségszerűen lezárul.

# 1. Projekt hierarchia

- Ideiglenes függőségi rendszer
  - Keresztbe vágja a szervezeti hierarchiát
  - Több céget (vevőt, szállítót) is érinthet
- Három szint
  - Irányító bizottság (steering committee)
  - Projekt menedzsment (management)
    - Egy v. több PM
    - Segítők: adminisztrátor, controlling, konfigurációs manager, stb.
  - Technikai szint
    - Team-ekbe rendeződve (azon belül kis hierarchia)
- A szinteken átível: minőségbiztosítás
  - Quality Assurance = QA
  - PM és/vagy technikai szinten

# Egy lehetséges hierarchia 3 cég részvételével



# 2. A projekt tervezése

- A terv részei
  - Szoftverfejlesztési terv
    - A vezér-dokumentum (Software Development Plan)
    - Résztvevők, alá-fölérendeltségek, felelősségek, dokumentumok, fő lépések, személyek kompetenciája és képzései, stb.
    - Elrettentésül:  
[http://sepo.spawar.navy.mil/SDP\\_Template.doc](http://sepo.spawar.navy.mil/SDP_Template.doc)
  - Ütemterv (tevékenységek, mérföldkövek dátuma)
    - Teendők, felelősök, határidők, összefüggések
    - Ütemezés és erőforrás diagram

***Részletesen csak ezekkel foglalkozunk***

# Projekt tervezés – további dokumentumok

- Minőségi terv
  - Minőségi követelmények (pl. elfogadható hibaszintek)
  - Benne: tesztelési terv (validációs terv), teszt procedúra
- Kockázatkezelési terv
  - Milyen technológiai, humán, gazdasági stb. veszélyek leselkednek a projektre?
  - Hogyan fognak ellenük védekezni?
- Konfiguráció-kezelési terv
  - Ki milyen részben felelős a források és a dokumentáció tárolásáért , a leszállítandó termékek elkészítéséért, teszteléséért, átadásáért?

***Ezek gyakran az SDP részeként készülnek el!***

# 3. A projekt ütemezése

- Tevékenységek, mérföldkövek, átadások ütemezése
  - Logikus sorrendben
  - Határidőre
  - Az emberek és más erőforrások rendelkezésre állása szerint
  - Lehetőleg az erőforrások egyenletes terhelésével
- Mindig van eltérés, de legyen minél kisebb
  - A projekt tervet menet közben is alakítani, pontosítani lehet és kell

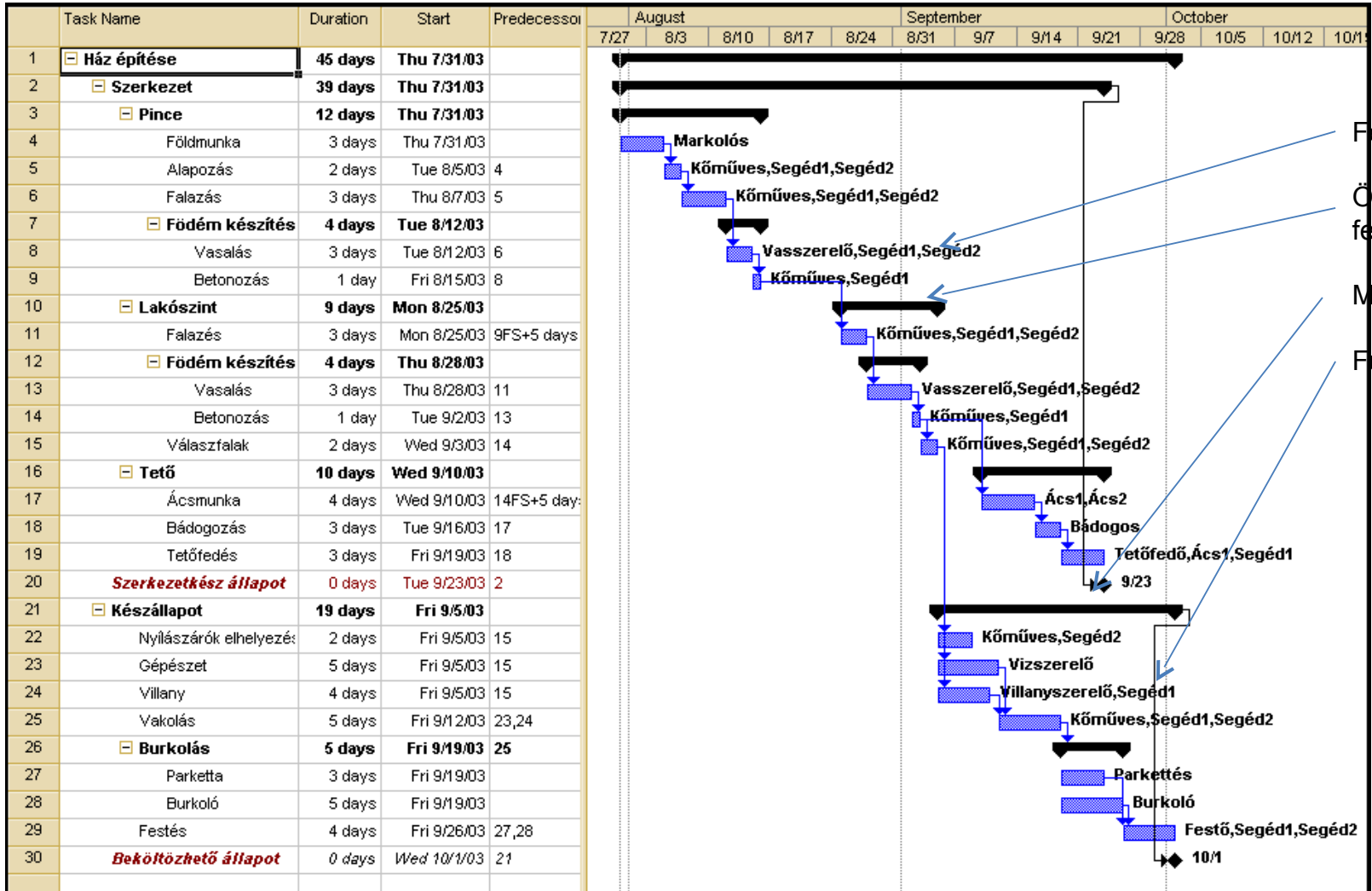
# Ütemezési fogalmak

- Tevékenység
- Mérföldkő (=milestone): a projekt egy jól azonosítható állapota
  - Ügyfél felé nincs feltétlenül szállítás
  - Pl.: „User interfész komponens tesztelésre kész”
- Szállítás (=delivery): (rész)eredmény átadása a projekten kívüli fogadónak (azaz az ügyfélnek)
  - Ezek egyúttal mérföldkövek is.
  - Pl.: „rendszerterv átadása”
- Erőforrás
  - Emberek, esetleg gépek, helyszínek
- Függőségek
  - Leggyakoribb a „Finish-to-Start”, azaz B A után kezdődhet

# Projektütemezési diagrammok

- Bevált grafikus, szemléletes módszerek
  - Egyszerű naptár
  - Háló-diagram
  - GANTT diagram
  - Erőforrás-diagram
- Jó ha kényelmes eszközök segítik a rajzolást, karbantartást

# GANTT Diagram



Feladat

Összetett feladat

Mérföldkö

Függőség

# GANTT Diagram

- Időtengelyen ábrázol
- A tevékenységeket időarányos sávok jelzik
  - Megjelölhetők az erőforrások és résztvevők
- Mérföldkövek, részeredmények
  - Kis rombusz
  - Technikailag a 0 hosszú tevékenységeket
- Jelezhetők a függőségek
  - Vég-kezdet, kezdet-kezdet stb.
- Azonosítható az időkritikus feladatok láncolata
- Bejelölhetők a készültségek, ill. az aktuális dátumhoz tartozó vonal
  - A vonalon cikk-cakkok jelzik az elmaradásokat és az előre elvégzett feladatokat.

# Erőforrás-diagram

Kinek mennyit feladata van az egyes napokon/hetekben?

	Resource Name	Work	Details	August				September				October		
				7/27	8/3	8/10	8/17	8/24	8/31	9/7	9/14	9/21	9/28	10/5
	+ Unassigned	0 hrs	Work											
1	+ Markolós	24 hrs	Work	16h	8h									
2	+ Kőműves	152 hrs	Work		32h	16h		24h	32h	16h	32h			
3	+ Segéd1	272 hrs	Work		32h	40h		40h	40h	32h	40h	24h	24h	
4	+ Segéd2	216 hrs	Work		32h	32h		40h	32h	16h	32h	8h	24h	
5	+ Vasszerelő	48 hrs	Work			24h		16h	8h					
6	+ Ács1	56 hrs	Work							24h	16h	16h		
7	+ Ács2	32 hrs	Work							24h	8h			
8	+ Bádogos	24 hrs	Work								24h			
9	+ Tetőfedő	24 hrs	Work								8h	16h		
10	+ Vízszerező	40 hrs	Work						8h	32h				
11	+ Villanyszerelő	32 hrs	Work						8h	24h				
12	+ Parkettés	24 hrs	Work								8h	16h		
13	+ Burkoló	40 hrs	Work								8h	32h		
14	+ Festő	32 hrs	Work									8h	24h	
			Work											

A táblázat automatikusan generálható a GANTT diagram alapján

Cél a (legalább szakaszonként) folyamatos, közel 100%-os kihasználtság

# 4. Microsoft Project

- Egy elterjedt, „könnyűsúlyú” projekt ütemező eszköz
- Az Office termékcsalád része, a többi eszközhöz hasonló interfésszel.
- Fogalmak
  - Feladatok: *tasks*
  - Mérföldkövek: *milestones*
  - Emberek (és más erőforrások): *resources*
  - Összefüggések: *dependencies*
- Funkciók
  - Ütemezés, munkák személyekhez rendelése
  - Követés, analízis, költségek követése

# ... vagy mégse...

- GNOME Planner!!!
  - A.k.a Mr.Projekt
- Még egyszerűbb
  - Szoftver projektekre kiváló
- Ingyenes
- Linuxon (is) fut
- Korlátai
  - Nincs attachment, megjegyzés
  - ???

# Projekt ütemterv készítésének lépései

- Feladatok felsorolása
  - Összetett és alábontott feladatok
- Emberek (erőforrások) felvitele
- Feladatok időigényének és egymásra épülésének meghatározása
- Emberek hozzárendelése a feladatokhoz
- Határidők és kihasználtság ellenőrzése
- Igazítás, optimalizálás

# Projekt követési funkciók

- Taszkok készültségének jelölése
- Progress line: az előrehaladottság ellenőrzése
- Feladat-listák generálása személyenként
  - Mikortól, meddig, mit, milyen csoportban
- Baselineing: a tervezett és megvalósuló projekt összehasonlítása.

# Vége!!!

Következnek az ellenőrző kérdések!!!

# Ellenőrző kérdések

1. Milyen célok motiválják a szoftverfejlesztési munka technológia-szerű megközelítését?
2. Mit nevezünk technológiának? Milyen előnyök várhatók akkor, ha valamit feladatot technológiaként végzünk?
3. Mik a szoftvertechnológiai eszközök? Sorolja fel a legjellemzőbb típusokat (5-6 db)!
4. Mik a szoftverfejlesztést közvetlenül meghatározó általános (az adott fejlesztési feladattól és követelményektől független) technikai tényezők?
5. Mi a CASE betűszó eredete?
6. Mi a fő akadálya annak, hogy Magyarország igazán sikeres és elismert szoftvergyártó környezet legyen?
7. Sorolja fel a jó szoftverek általános tulajdonságait (10 db).
8. Mik egy szoftverfejlesztési munka leszállítandó eredményei (3 féle).
9. Van-e alternatívája a fejlesztőnként több ezer dolláros szoftvertechnológiai szoftvereknek? Soroljon fel min. 3 ilyen megoldást, megjelölve a feladatot, amire használható.
10. Mi egy futtatókörnyezet, általában mit várhatunk tőle? Mik tekinthetők minimál-szolgáltatásnak?

# folyt.

11. Mik a kb. 15 éve, ill. a most jellemző futtatókörnyezet szolgáltatások?
12. Soroljon fel „prémium kategóriás” futtatókörnyezeteket (4 db)!
  
13. Mik a projekt jellegű munkavégzés általános ismérvei? Hogyan nevezzük a nem-projekt keretében végzett feladatokat?
14. Felelősségük, szerepük szerint hogyan csoportosíthatók egy nagyobb projekt résztvevői? (4 csoport).
15. Miért fontos a projekt minőségbiztosítás viszonylagos függetlensége?
16. Nagy projekteknél milyen dokumentumokat szokás a projekt indulásakor elkészíteni (5 db.)
17. Milyen tényezőket rögzít a szoftverfejlesztési terv?
  
18. Milyen fogalmakat használunk a projekt ütemezés elkészítésénél? Azaz: milyen fogalmakat kezel egy projekt ütemező szoftver?
19. Mik a cél-szemponatok a projekt ütemezésénél?
20. Mi az ütemezés kritikus útvonala?
21. Miket, milyen grafikai szimbólummal ábrázolhatunk a GANTT diagramon?
22. Planner (v. Microsoft Projekt) használata esetén mik egy szoftverfejlesztési projekt ütemezési terv elkészítésének, hangolásának lépései?