

Üdvözli Önöket A Programozás Technológiai Eszközei c. tantárgy!!

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

IKKK

Bakay Árpád dr.

NETvisor kft

(30) 385 1711

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
- Egyetemek: BME, Vanderbilt (USA), NYME (Sopron)

NETvisor

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

Miről lesz szó?

- A professzionális szoftverfejlesztést támogató eszközökről
 - A módszertanok már ismertek számunkra
 - Kurrens, korszerű eszközöket tanulunk
 - Cél az eszközök gyakorlati használatának elsajátítása
- Korszerű szoftver architektúrák
 - J2EE alkalmazáserver
 - Végig egyetlen viszonylag komplex alkalmazással foglalkozunk, ezen dolgozgatunk
- *Csak ami a gyakorlatban is hasznos!!!*

Általános információ

- Syllabus
 - website:
<http://people.inf.elte.hu/bakaya/progtechgyak2.html>
 - email-lista?
- Órák:
 - Szerda 16:00:00-17:30:00
 - Összesen 12 alkalom
 - április 4: nincs óra (11-én van!)
 - május 16 az utolsó óra, vizsga ZH 4 órában!!

Általános információ – folyt.

- Könyvek:
 - Sommerville - Szofver rendszerek fejlesztése
 - Panem kiadó 6900 ft
 - McConnell: Code Complete
 - magyarul??, (Amazonon 30 USD)
 - J2EE útikalauz Java Programozóknak
 - <http://java.inf.elte.hu>, max. 6000 ft
 - Objektum elvű modellalkotás UML-ben
 - Varga L., Sike S.
 - Java tutorialok: java.sun.com
 - Link gyűjtemény

Számonkérés

- Kis ZH-k: rendszeresen, 6-8 db
- Házi feladat: 2 részletben
- Nagy vizsga-ZH

- „Honor code”, szankciók



Munkához!!!



1. Programozás és technológia

- Technológia: egy termelési folyamat elmélete és gyakorlata
 - Üzemszerű „szoftver-gyártás”
 - Szisztematikus, szabályzott, ellenőrzött
 - A legmegfelelőbb eszközökkel
 - Költség-, minőség-, idő-optimalizálás.
- Miért kell mindez a szoftverfejlesztéshez?
 - A bonyolultság miatt az ösztönös megközelítés a elérte lehetőségek határát
 - A csapatmunka, és a hosszabb távú karbantarthatóság bizonyos szabályok betartását igényli.
- *Magyarországon ez (még) nem az általános megközelítés*

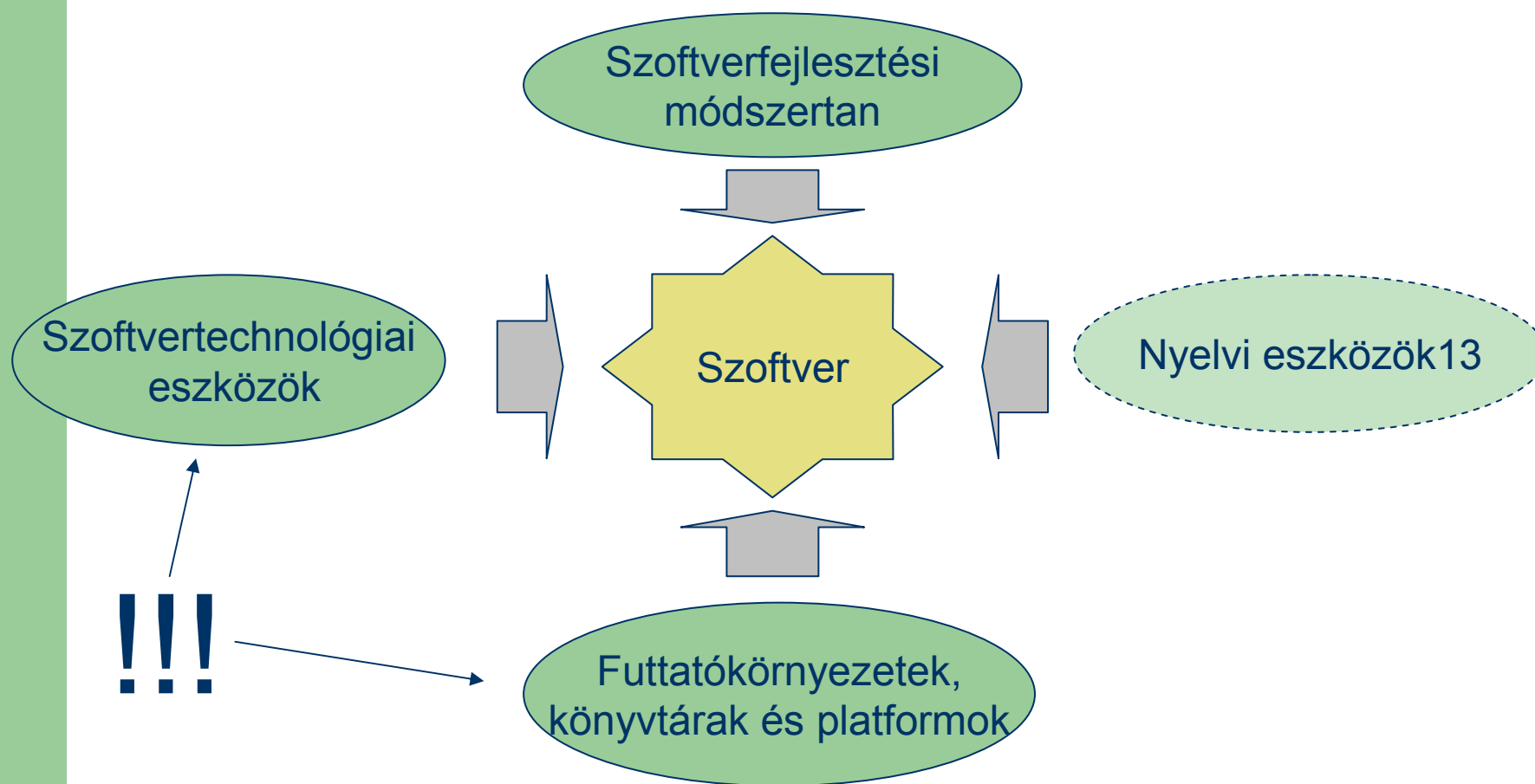
Szoftvertchnológiai eszközök

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

Computer Aided Software Engineering – CASE

...és ami nem CASE: compilerek, szövegszerkesztők, rajzolóprogramok...

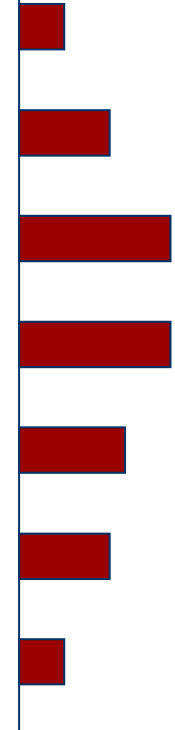
A szoftver fejlesztését befolyásoló adottságok



Szoftvertchnológiai eszköz-kategóriák

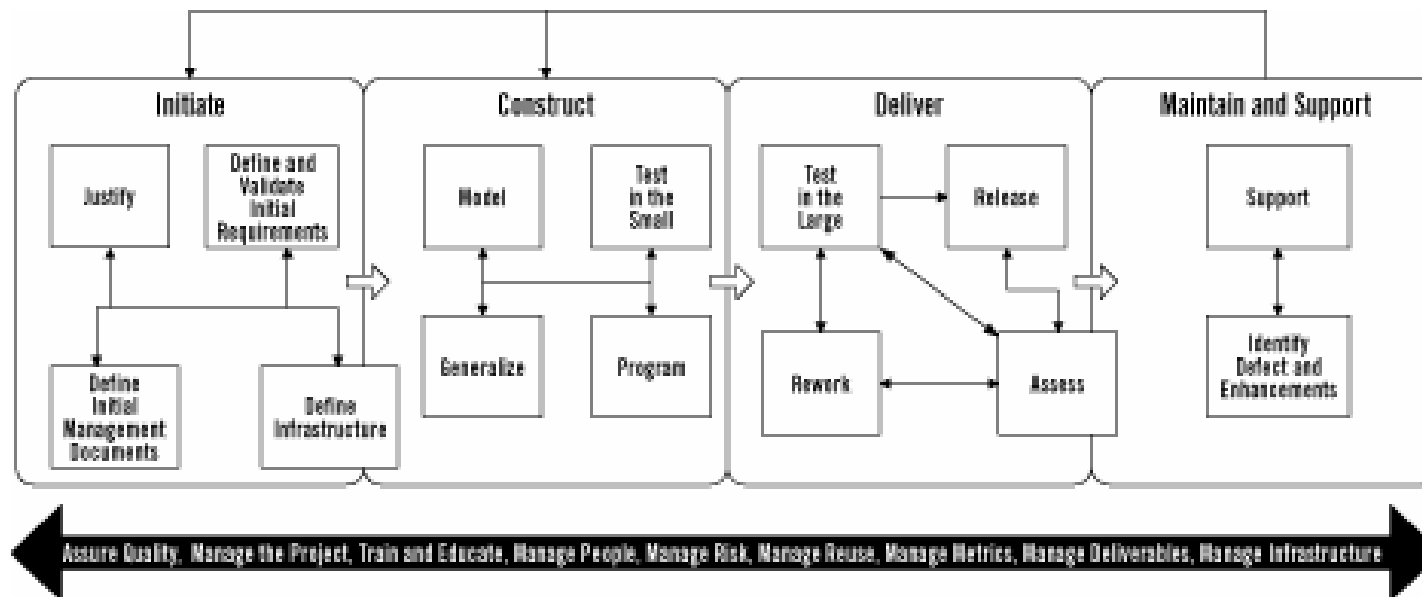
- Projekt tervezés és irányítás
- Követelmény-elemzés
- Tervezés
- Fejlesztés és hibajavítás
- Tesztelés
- Konfiguráció-, változás- és feladatkezelés
- Dokumentáció generálása

Hangsúly
nálunk



Más nézet -- RUP

Object Oriented Software Process



The Object-Oriented Software Process (OO SP)

"Serial in the large, iterative in the small, delivering incremental releases over time."

Eszközök piaci kínálata

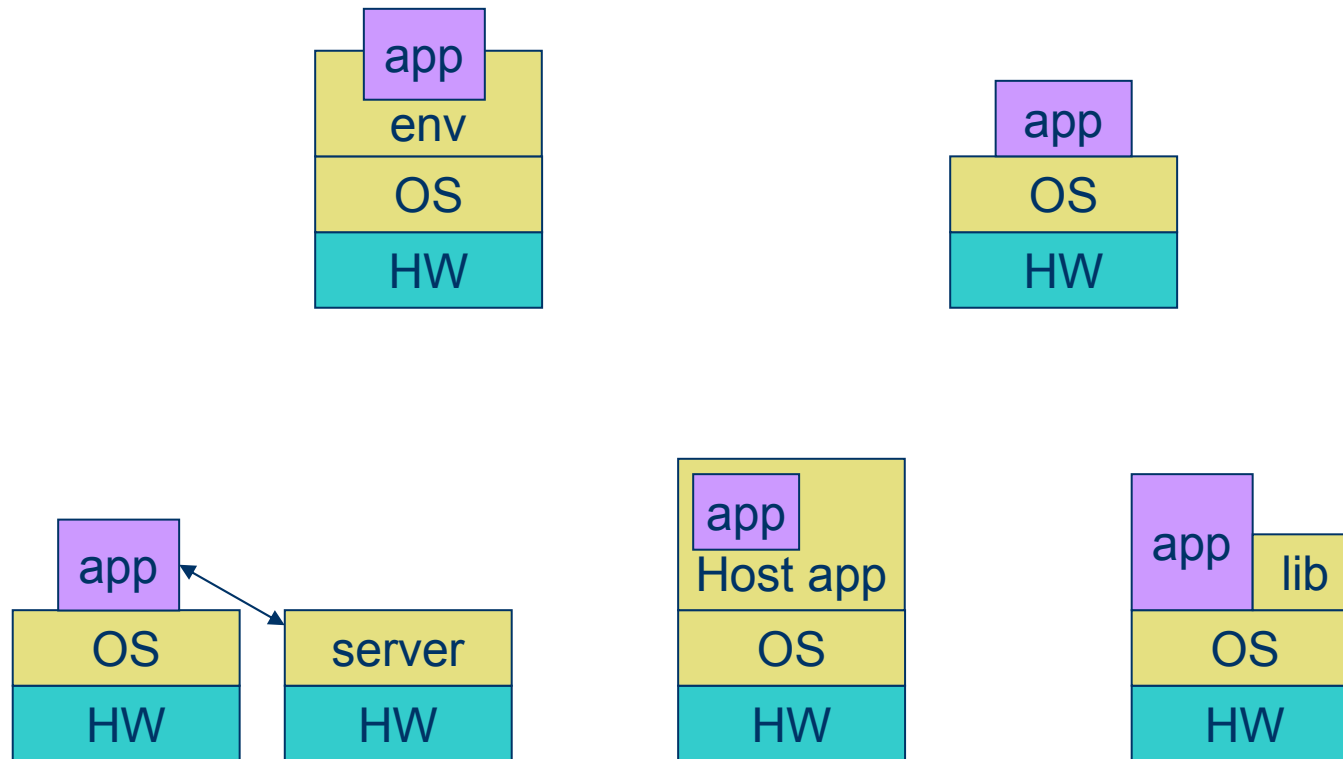
(korántsem teljes!!)

	Rational/IBM	Telelogic	Borland	Microsoft	FREE!!!
Projekt Management				MS Project	
Követelménykezelés	RequisitePro	Doors	Caliber RM		
Modellezés	Rose	Tau	Together		
Konfig. management	ClearCase	Synergy	StarTeam	Visual SourceSafe	CVS Subversion
Feladatkezelés	ClearQuest	Synergy	StarTeam		Bugzilla BugTracker
Tesztelés	TestSuite				xUnit
Dokumentáció	SoDa				javadoc

Futtatókörnyezetek

- A programjaink csaknem minden esetben egy meglévő szoftverkörnyezetben futnak, és kihasználják annak szolgáltatásait
 - Op. rendszer
 - Pl. assembly programoknál
 - Nyelvi futtatókörnyezet
 - Pl. Basic, Logo interpreterek szolgáltatásai
 - Rendszerbe installált library-k
 - Pl. Unixon futó C/C++ programok
 - Host program
 - Pl. VB for MS Office, pluginok
 - Frameworkok
 - Kész felhasználható építőelemek
 - *Sőt: adatbázis, stb. szerver-környezetek*

Modellek azonosítása



Röviden a futtatókörnyezetekről - jellemző szolgáltatási szintek

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

(folyt)

- „Premium” (Corba, COM+, J2EE, MS .NET)
 - tranzakció-kezelés,
 - persistence, caching
 - security
 - encryption
 - AAA (authenticate, authorize, audit)
 - single sign-on
 - load-balancing, failover,
 - messaging
 - configuration, diagnostics, management
 - logging

2. A szoftverfejlesztési projekt

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

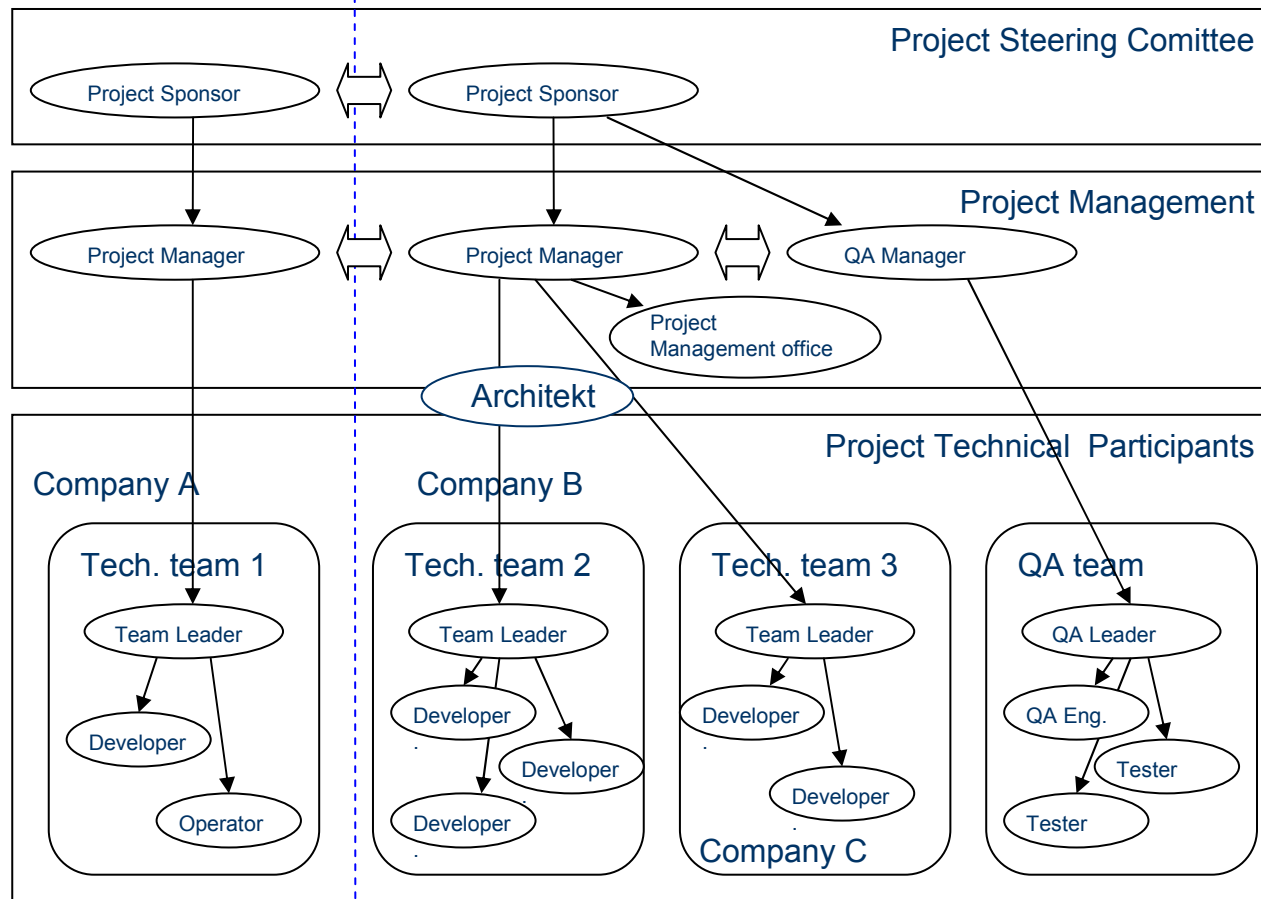
Ha projekt, akkor...

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

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
 - PM vagy technikai szinten

Egy lehetséges hierarchia



A projekt tervezése

- A terv részei

- Szoftverfejlesztési terv
 - A vezér-dokumentum
- **Ütemterv (tevékenységek, mérföldkövek dátuma)**
 - Ütemezés és erőforrás diagram
- Minőségi terv
 - Benne: tesztelési terv (validációs terv)
- Kockázatkezelési terv
- Konfiguráció-kezelési terv

Csak ezzel foglalkozunk

A projekt ütemezése

- Tevékenységek, mérföldkövek, részeredmények 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

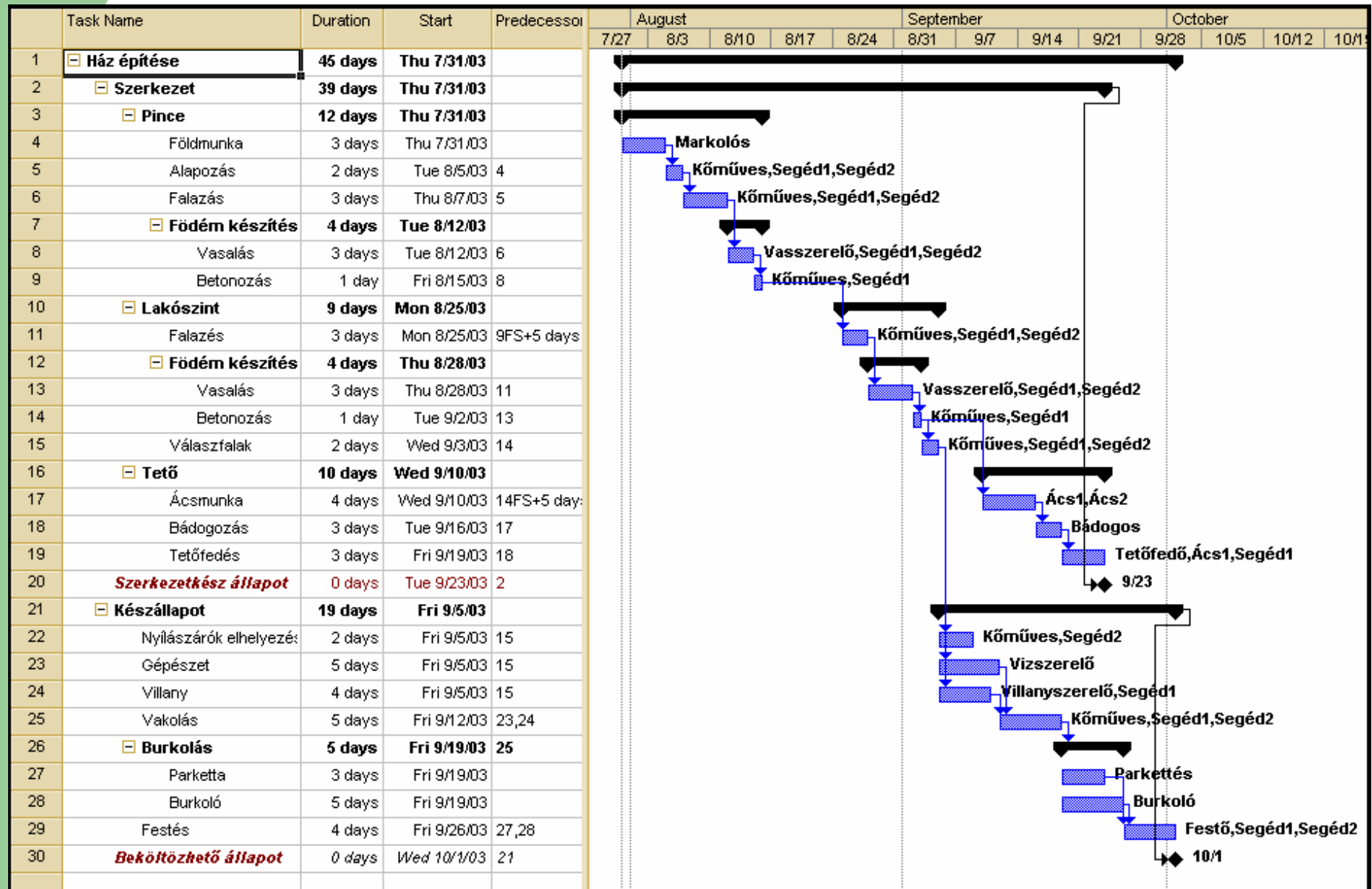
Fogalmak

- **Tevékenység**
 - Szereplők, időigény, időzítés
- **(Rész)eredmény leszállítása (delivery): amikor az ügyfél valamit megkap**
 - Pl.: „rendszerterv átadása”
- **Mérföldkő: a projekt egy jól azonosítható állapota**
 - Ügyfél felé nincs feltétlenül szállítás
 - Pl.: „User interfész komponens tesztelésre kész”

Projektütemezési diagrammok

- Grafikus, szemléletes módszerek
 - Naptár
 - Háló-diagram
 - GANTT diagram
 - Erőforrás-diagram
- Jó ha kényelmes eszközök segítik a rajzolást, karbantartást

GANTT Diagram



GANTT Diagram

- Időtengelyen ábrázol
- A tevékenységeket időarányos sávok jelzik
 - Megjelölhetők az erőforrások és résztvevők
- Mérföldkövek, részeredmények: kis rombusz
- Jelezhetők a függőségek
- Időkritikus feladatok láncolata

Erőforrás-diagram

	i	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		+ Vizszerelő	40 hrs	Work						8h	32h				
11		+ Villanszerelő	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											

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

Szoftver projekt jellemző fázisai

- (Ajánlati szakasz)
 - Rendszerint a projekt előtt
- Specifikáció 20%
- Tervezés 25%
- Implementáció 25%
- Validáció és integráció 30%
- (Karbantartás, evolúció)
 - Rendszerint a projekt után

Mekkora egy szoftver projekt?

- **Kicsi:** < 1 emberév, kb. 3 hónap időtartam
- **Nagy:** > 20 emberév, minimum 1 év időtartam
- **Közepes:** ami közte van

A szoftverfejlesztési projekt jellemzői

- Egyszerűbb mert:
 - Mérsékelt anyag és eszközfelhasználás, ill. logisztika
 - A résztvevők tudása jórészt konvertálható
 - Költségek zömét a fix munkabér teszi ki
- Bonyolultabb, mert
 - Az egyes fázisok munkaigénye nehezen becsülhető
 - A szoftverfejlesztőknek érzékeny személyisége

A szoftver projekt terv dinamikusan változik

- Igazítani kell
 - Elakadások, késések
 - Új/bővülő/szűkülő feladatok
 - Más projektek hatása az erőforrásokra
- Nem csodaszer, csak segédeszköz

3. Ursula - A mi projektünk

- Orvosi nyilvántartó rendszer
- Egyelőre egy előzetes (informális) specifikációnk van
- Projekt ütemezés a részletes specifikáció előtt szükséges
 - Hogy pl. lehessen pénzügyi ajánlatot tenni
 - Ennek része a precíz specifikáció készítése!!!!

Ellentmondás???
Igen!!!

Ursula informális specifikáció 1. rész

Általános ismertetés

Az alkalmazás betegek, kórtörténetek és különféle kezelések (vizsgálat, laborvizsgálat, műtét, rutinellátás, kontroll) nyilvántartására szolgál. Az alkalmazással követhető a betegségek lefolyása, diagnózisa, az egyes kezelések időpontja és főbb paraméterei, valamint az egyes orvosok és kórházi osztályok tevékenysége, teljesítménye.

Ursula informális specifikáció 2. rész

Funkcionális leírás

Az alkalmazás felhasználói, és a számukra elérhető főbb műveletek a következők:

- Beteg: saját adatok megtekintése, bejelentkezés kezelésre, időpont-módosítás
- Adminisztrátor: új beteg és új eset felvétele, bejelentés kezelésre.
- Orvos: kezelés adatainak kitöltése, új elvégzendő kezelés rögzítése
- Labor: az elvégzett laborvizsgálat eredményének felvitele
- Kórházi adminisztráció: új orvos, osztály felvétele, orvosok osztályhoz rendelése, lekérdezések az orvosok és az osztályok munkájáról

Ursula informális specifikáció 3. rész

Egyéb követelmények

- A UI legyen Web-alapú – *UI technológia*
- A rendszer kapacitása legyen elegendő 300 kórházi dolgozó és 5000 beteg, 10000 eset és 100000 kezelés nyilvántartására. -- *hatékonyság*
- Az elvárt teljesítmény a fenti maximális kapacitásnál is legyen max. 3 sec várakozási idő képernyőként (kivéve a statisztikai, összesítési funkciókat, ahol hosszabb várakozás megengedett). -- *hatékonyság*
- Az alkalmazás valamilyen kereskedelmi relációs adatbáziskezelőben tárolja az adatokat. -- *karbantarthatóság, teljesítmény, biztonság*
- Az alkalmazás minden felhasználótól követelje meg a jelszavas bejelentkezést. -- *biztonság*
- Az alkalmazás legyen könnyen bővíthető (pl. diagnosztikai szakértői rendszer, előjegyzési naptár funkciók, betegek értesítése emailben, stb.) -- *bővíthetőség*
- Legyen lehetőség más rendszerekből való automatikus adatcserére (pl. laborszámítógép, központi pénzügyi rendszer, személyzeti rendszer, PalmPilot stb.) -- *bővíthetőség*

Ursula informális specifikáció 4. rész

Határidők

A rendszert 2006 májusában kell üzembe helyezni.

Milyen a 'Jó minőségű' szoftver?

- általános elvárások

- A felhasználói igényeknek jól megfelel
 - könnyű kezelhetőség vs. sokoldalúság
- Hatékony
- Üzembiztos
 - Pl. rossz szokatlan adata is
- Továbbfejleszthető
- Karbantartható
 - Üzemeltethető (pl. monitorozás, log fileok)
- Komplettn:
 - Funkcionalitás, dokumentáció, adminisztrációs funkciók
- ***Határidőre elkészül!!!***

Összefoglalás

- Szoftvertechnológiai eszközök
 - Definíció, funkcionális osztályozás, hangsúlyok
- A szoftver projekt
 - Szervezet, tervezés, dokumentumok
 - Ütemezés, diagram-típusok
- Példaalkalmazás első lépése - előzetes specifikáció

Vége!!!

Köszönöm a figyelmet!!!