



Software Engineering



Software Engineering

- Software Engineering értelmezése
 - Az a folyamat, mely eredményeként létrehozunk egy adott feladatot megvalósító szoftver rendszert.
 - Tevékenységek, technológia, módszerek, eszközök...
 - Számítógép alapú rendszert hozunk létre



Engineering

- Mérnöki munka rendszerek létrehozására (~ fejlesztés)
 - Építő mérnök
 - Gépész mérnök
 - Villamos mérnök
 - Szoftver mérnök
- Modellezés – általános eszköz
 - leendő rendszer leírása, specifikációja, terve



Engineering

- System Engineering („rendszer”...)
 - Business process engineering (üzleti folyamat...)
 - vállalat működésének, üzleti folyamatainak tervezése, szervezése
 - modellezzük az üzlet működését, környezetét
 - információ feldolgozás lesz a középpontban
 - nem koncentrálnak kizárólag a rendszerben használandó szoftverre
 - Product engineering (termék...)
 - termék tervezése
 - modellezzük a terméket, annak használatát
- Software Engineering (szoftver...)
 - szoftver alkalmazásokat, eszközöket ad a SE által definiált feladatok megoldására
 - a szoftver rendszer (alkalmazás) létrehozására koncentrálnak



Software Engineering

Lépések definíciója, technikai tartalma



Szoftver fejlesztés – lépések

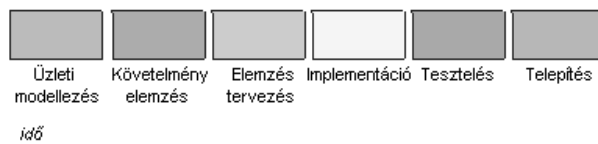
- (Üzleti modellezés, Termék modellezés)
- **Követelmény (feldolgozás, elemzés.)**
- **Elemzés**
- **Tervezés**
- **Implementáció**
- **Tesztelés**
- **Telepítés**
- (Karbantartás, követés, továbbfejlesztés)

Szoftver fejlesztés lépései

- Fejlesztési tevékenységek
 - szoftver előállítás érdekében tett műveletek (pl. modellezés, kódolás, tervezés stb.)
- Fejlesztési termékek:
 - fejlesztési lépések eredményei
 - pl. modellek, kód, dokumentáció stb.

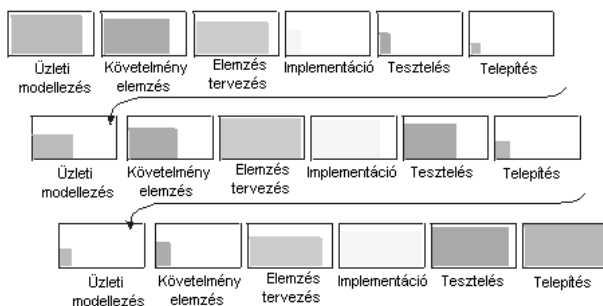
Fejlesztés lépéseinek sorrendje

- Vizesés modell szerint



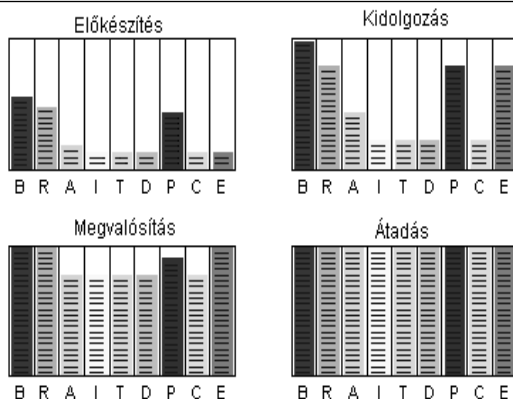
Fejlesztés lépéseinek sorrendje

○ Iteratív fejlesztési modell szerint



Fejlesztési termékek előállítása a fejlesztés során

Iteratív fejlesztés:
Az iterációk során egyre több termék áll elő, és a termékek érettsége egyre nő.



- B : Üzleti modellezés termékei
- R : Követelmény-elemzés termékei
- A : Elemzés-tervezés termékei
- I : Implementáció termékei
- T : Tesztelés termékei
- D : Telepítés termékei
- P : Projektvezetés termékei
- C : Konfiguráció és változás-kezelés termékei
- E : Környezet kialakítása termékei



1. System Engineering



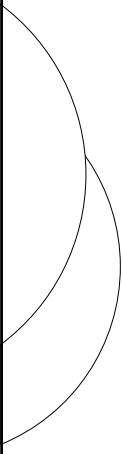
System Engineering (SysEng)

- Általános értelemben vett rendszer fejlesztés
- System Engineering lehet
 - **Business Process Engineering**
ha vállalat (működésének) (át)szervezésével foglalkozunk
Pl.: számlázási rendszer,...
 - **Product Engineering**
ha egy termék előállítása a cél
pl.: mobil telefon, repülőgép vezérlő rsz.



Számítógépes rendszerek

- Olyan rendszer, mely elemei információt feldolgozva dolgoznak egy előre definiált cél megvalósítása érdekében.
- Részai:
 - szoftver
 - hardver
 - emberek (felhasználók, operátorok)
 - dokumentáció (rendszer leírása)
 - eljárások (a rendszer használatának lépései)



1.1. Business Process Engineering (BPE)

Business Process Engineering (BPE)

- Célja:
 - Olyan felépítés (architektúrák) definiálása az üzleti tevékenység számára, mely lehetővé teszi az információ hatékony felhasználását és ezáltal az üzlet hatékonyabb működését.
- Miért van rá szükség?
 - nagy szervezetek (vállalatok)
 - összetett információ feldolgozási lépések
 - szükséges a szervezet működését leírni modellezni (megérteni), optimalizálni az adott cél érdekében
 - megfelelő számítógépes (szoftver) rendszert készíteni a működés támogatására

BPE során megvalósítandó feladatok

- Leírni, analizálni és megtervezni az üzleti folyamatokat
- Szempontok:
 - **adat felépítés (architektúra)**
 - üzleti folyamatok által kezelt (feldolgozott) adatok
 - **applikációs felépítés (architektúra)**
 - adatokat feldolgozó elemek működése (szoftverek, emberek)
 - **technológiai felépítés (architektúra)**
 - szoftver, hardver elemek, melyek az adatok tárolását és az applikációk működését támogatják



BPE lépései (szintjei) I.

- **Information Strategy Planning (ISP)**
 - Információs stratégia tervezés
 - üzleti területek (domain-ek) meghatározása (pl. gyártás, marketing, pénzügyek stb.)
- **Business Area Analysis (BAA)**
 - Üzleti terület elemzése
 - Adott területen szükséges adatok és funkciók definiálása, valamint az együttműködés más területekkel.
 - Eredményeként meghatározható, milyen (információs rendszer) számítógépes rendszer alkalmazható az adott területen.



BPE lépései (szintjei) II.

- **Business System Design (BSD)**
 - Üzleti rendszerek tervezése
 - → Software engineering
 - Követelmények részletes és pontos meghatározása az alkalmazandó (információs rendszer) számítógépes rendszer számára.
- **Construction and Integration (C&I)**
 - Megvalósítás és integrálás
 - Alkalmazás megvalósítása, integrálása az adott környezetbe.



1.2. Product Engineering



Product Engineering

- Célja olyan működő termék létrehozása, mely kielégíti a felhasználó követelményeit.
- Cél a rendszer architektúrájának, felépítésének meghatározása (megtervezése)
- Felépítés (architektúra) a következő szempontokból:
 - szoftver
 - hardver
 - adatok (adatbázis)
 - emberek (felhasználók, operátorok)



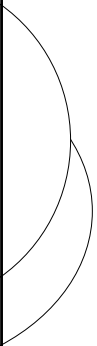
Product Engineering lépései

- Requirement Engineering
 - Követelmény (tervezés)
 - Rendszerre vonatkozó követelmények meghatározása:
 - funkció, viselkedés, teljesítmény, tervezési, interfész megkötések stb.
- System Component Engineering
 - Komponens tervezés
 - → Software engineering
 - analízis, tervezés, megvalósítás, integrálás
 - Hardver tervezés
 - Emberi erőforrás tervezés



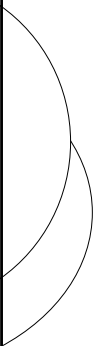
1.3. Requirement Engineering

Követelmény (meghatározás, specifikálás, elemzés... ?)



Requirement Engineering (Követelmény...)

- A System Engineering lépések eredménye (kimenete):
 - megvalósítandó rendszer specifikációja
- Kérdés:
 - a specifikáció tényleg olyan rendszert ír le, ami kielégíti a felhasználó igényeit?
- Megoldás:
 - Requirement Engineering definiál olyan lépések sorozatát, ami ezt biztosítja.



Requirement Engineering (RE) (Követelmény...)

- A RE-ről beszélhetünk a SysEng részeként, és a Software Engineering részeként
- A System Engineering részeként a RE nem csak a szoftver rendszerre vonatkozó követelményekkel foglalkozik
- A RE lépései, a megvalósítandó feladatok hasonlóak mind a két esetben, a módszerek nem feltétlenül



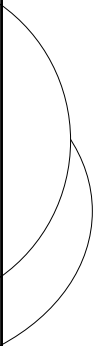
Requirement Engineering lépései

- Követelmények kiderítése, összegyűjtése
- Követelmények elemzése és egyeztetése
- Követelmény specifikáció készítése
- Rendszer modellezés
- Követelmények validálása
- Követelmények kezelése, követése



Követelmények kiderítése, összegyűjtése

- Feladatok:
 - (scope)
 - megoldandó probléma határainak definiálása
 - (understanding)
 - tudatosítani, milyen lehetőségei vannak a rendszernek, triviális és félreérthető követelmények elhagyása, ellenőrizhető követelmények definiálása
 - (volatility)
 - követelmények állandósága/változékonysága



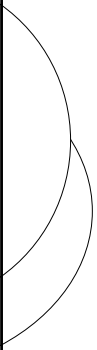
Követelmények elemzése és egyeztetése

- kategorizálás, követelmény típusok definiálása
- követelmények közötti összefüggések:
 - konzisztencia,
 - prioritások, ...
- különböző forrásból származó (stakeholder) követelmények közötti egyeztetés
- követelményekhez kapcsolódó (megvalósításhoz) kockázatok tisztázása



Követelmény specifikáció készítése

- követelmények rögzítése valamilyen formalizált módon
- konzisztens és érthető leírása a követelményeknek
- különböző formában lehetséges:
 - szöveges (template alapján), grafikus, matematikai modell, prototípus, stb...
- a specifikáció kötelező része a rendszer bemeneteinek és kimeneteinek definiálása
- **Eredmény: Rendszer Specifikáció**
 - alapja minden további mérnöki munkának (szoftver, hardver, emberi erőforrás, stb.)



Rendszer modellezés

- a teljes rendszer leképzése egy modellben
 - pl.: ház tervrajza
- az egyes követelményekben leírt tulajdonságok egy modellben történő leírása
- követelmények közötti kapcsolat átlátható reprezentálása
- Szabványos technika (pl.):
 - bemenet feldolgozás, kimenet feldolgozás, felhasználói interfész, irányító mag, szerviz...



Követelmények validálása

- Követelmény halmaz
 - teljes-e,
 - ellentmondás mentes-e,
 - hiba mentes-e?
- Technikák:
 - review
 - szimuláció
- Általános követelményekre vonatkozó kérdések:
 - érthető, jól definiált a követelmény?
 - tesztelhető?
 - forrása ismert?
 - sért-e valamilyen szakterületi követelményt?
 - ...



Követelmények kezelése, követése

- Követelmények változhatnak a rendszer élete során. Változásokat követni kell.
- Összefüggések követelmények és a rendszer különböző részei között:
 - rendszer funkciók és követelmények
 - interfészek és követelmények
 - részrendszerek és követelmények
 - ...

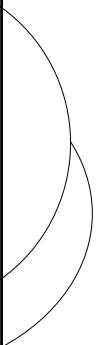


2. Elemzés – Software requirement engineering



Elemzés

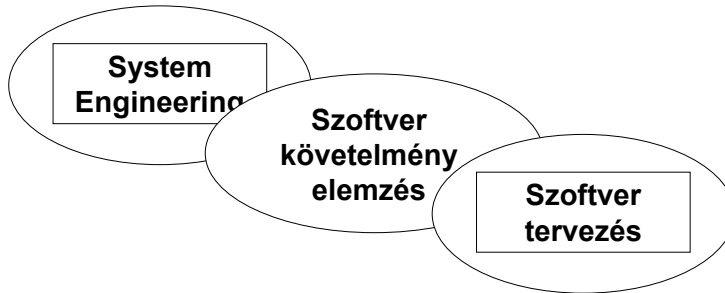
- Software engineering keretében végzett Requirement Engineering
- Szigorúan a létrehozandó szoftver rendszerre vonatkozó követelményekkel foglalkozik
- Elemzés ~ Software requirement engineering



Követelmény elemzés szerepe a sw fejlesztésben

- System Engineering (SysEng) → Követelmény elemzés → Szoftver tervezés
- A (SE során specifikált) követelményekből a megvalósítandó rendszerre vonatkozó részletes követelmények meghatározása
- Interaktív: felhasználó + elemző
- Követelmény elemzés:
 - MIT kell csinálni!!
 - NEM HOGYAN kell csinálni!!

Elemzés kapcsolata a szoftver rendszer fejlesztés lépéseivel



Követelmény elemzés általános lépései

- probléma azonosítása
 - követelmények összegyűjtése
 - rendszer határainak tisztázása...
- kiértékelés és szintézis
 - konzisztencia
- modellezés
- specifikáció készítés
- review (értékelő áttekintés)

2.1. Követelmények kiderítése, összegyűjtése szoftver fejlesztéskor

- interaktív rész (beszélgetés, interjú, stb.)
- cél:
 - kideríteni a „résztvényeseket” – kik húznak hasznot a rendszerből (felhasználók, megrendelő, stb.)
 - igények: funkció, kimenet, teljesítmény...
 - megfelelő embert kérdeztünk, van-e aki jobban tudja definiálni

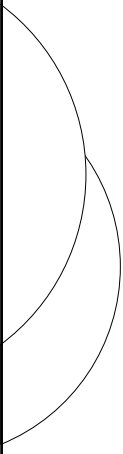
Technikák követelmények gyűjtésére

- Facilitated Application Specification Technique (FAST)
 - Felhasználókat és sw fejlesztőket egy oldalra gyűjtöm
 - formalizált lépések a követelmények definiálására a két oldal összehangolásával
- Quality Function Deployment
 - követelményeket kategorizálom:
 - közvetlenül definiált alap követelmények
 - elvárt, implicit követelmények
 - izgalmas követelmények (az a bizonyos plusz)

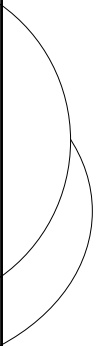


Technikák követelmények gyűjtésére - Use-Cases

- aktor
 - rendszerrel kapcsolatba kerülő felhasználó v. rendszer
 - valamit „akar” a rendszertől: input, output
 - szerepkört definiál!
- use-case
 - rendszer használata adott cél érdekében
 - rendszer használatakor történő események leírása
 - forgatókönyv, aktivitás diagram...

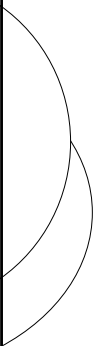


2.2. Elemzési alapelvek, módszerek, technikák (gy)



Elemzési alapelvek, módszerek technikák (gy)

- Elemzéskor használt alapelvek:
 - le kell írni, ill. megérteni a probléma megoldásához szükséges információ szerkezetét (information domain)
 - definiálni kell a szoftver által megvalósítandó funkciókat (functional domain)
 - definiálni kell a szoftver viselkedését (külső eseményekre adott válaszok) (behavioral domain)
 - a szoftvert leíró modelleket (információ, funkció, viselkedés) strukturálni kell
 - az elemzés menete során a alapvető tulajdonságoktól kell a megvalósítási részletek



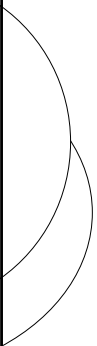
Information domain (gy) Információs domén

- sw alkalmazás ~ adat feldolgozás, konverzió
 - bemenet → kimenet
 - adat + vezérlő információ
- Information domain tartalma (nézetek)
 - rendszer által kezelt adattartalmak és azok kapcsolata
 - információ áramlás leírása a rendszerben
 - információ struktúrája (adat elemek és vezérlő elemek kapcsolata, felépítése stb.)



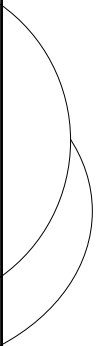
Modellezés (gy)

- funkciót leíró modell (functional domain)
- viselkedést leíró modell (behavioral domain)



Részekre bontás (gy)

- komplex probléma → egyszerűbb rész problémákra
- horizontális partícionálás
- vertikális partícionálás



Alapvető és implementációs nézet (gy)

- A szoftver követelmények lényegi, alapvető nézete (~logikai nézet):
 - nem a megvalósítás (implementáció) érdekli
 - megvalósítandó funkciók és a feldolgozandó adatok leírása
- A szoftverkövetelmények implementációs nézete (~fizikai nézet)
 - a végleges implementáció érdekli
 - megvalósítandó funkciók és a feldolgozandó adatok tényleges megvalósításának leírása
 - gyakran csak tervezéskor készül el, de néha korábban kell (pl.)



2.3. Prototípus készítés



Prototípus készítés

- prototípus készítési megközelítések
 - eldobandó prototípus
 - továbbfejlesztendő prototípus
- módszerek és eszközök
 - negyedik generációs technikák (4GT)
 - DB lekérdezés, interfész, grafikus felület stb...
 - újrahasználható szoftver komponensek
 - formális specifikáció + prototípus generáló környezet:
 - szöveges specifikáció
 - automatikus kódgenerálás
 - követelmények finomítása



2.4. Szoftver specifikáció



Szoftver specifikáció

- alapelvek, általános szabályok a specifikáció készítésre vonatkozóan:
 - funkció és az implementáció szétválasztása
 - rendszer viselkedésének (funkció, adatkonverzió) leírása külső események hatására
 - ...
- Eredmény általában szöveg és modell, ami lehet elektronikus, ill. írott formában.



Szoftver specifikáció eredménye

Szoftver Követelmény Specifikáció
Software Requirement Specification (SRS)

- SRS kötelező részei:
 - Bevezető – probléma definiálása
 - Információ leírása (← információs domain)
 - Funkció leírása (← functional domain)
 - Viselkedés leírása (← behavioral domain)
 - Rendszer (v. funkció) validációjának lehetséges módja (teszteléshez!!)



2.4. Szoftver review

- SRS
 - teljes, konzisztens, pontos ???
 - általában „manuálisan”
 - nehéz!!
 - prototípus → működtetés
 - modell → szimuláció
 - megrendelővel jóváhagyni
 - a szerződés alapja 😊



3. Elemzés módszerei

3. Elemzés módszerei

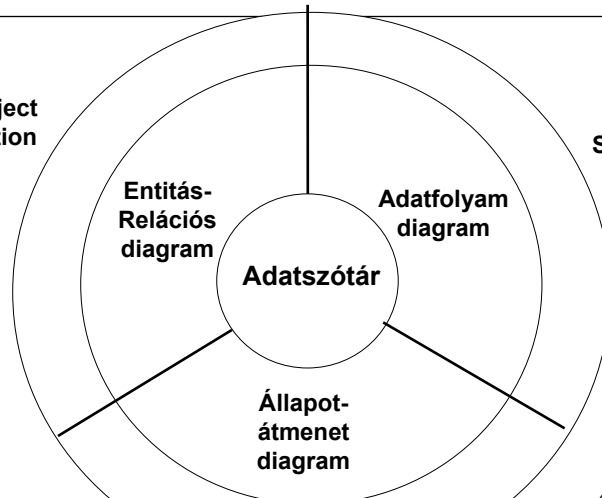
- Elemzés (teljes fejlesztés) megvalósítási alternatívái
 - strukturált → procedurális program fejlesztés
 - objektum orientált → objektum orientált rendszer fejlesztése

Strukturált elemzés során készített modell szerkezete

Data Object
Description

Process
Specification

Control
Specification



© C



Strukturált elemzés eszközei

- Adat könyvtár (dictionary)
 - Rendszer által kezelt adatok leírása
- Entitás relációs diagram (entity relations.)
 - Adat objektumok és az adatelemek egymáshoz való viszonyának leírása
- Adat folyam diagram (dataflow)
 - Adat transzformáció és adat mozgás, valamint az adat manipuláló funkciók leírása
- Állapot átmeneti diagram (state trans.)
 - Vezérlés leírása



„ Strukturált elemzés” FEJEZET TÖBBI RÉSZE A GYAKORLATON VOLT !!!

- ER diagram
- Adat folyam diagram
- Állapot átmenet diagram
- ...



4. Tervezés



Tervezés

- Cél: tervezési modell készítése
- Tervezési modell
 - direkt módon megvalósítható rendszer elemek leírása
- Tervezés „fázisai” (Belady)
 - divergálás – alternatívák
 - konvergálás – választás
 - kreatív folyamat!! → döntések!
- Alternatívák:
 - Strukturált tervezés
 - Objektum orientált tervezés



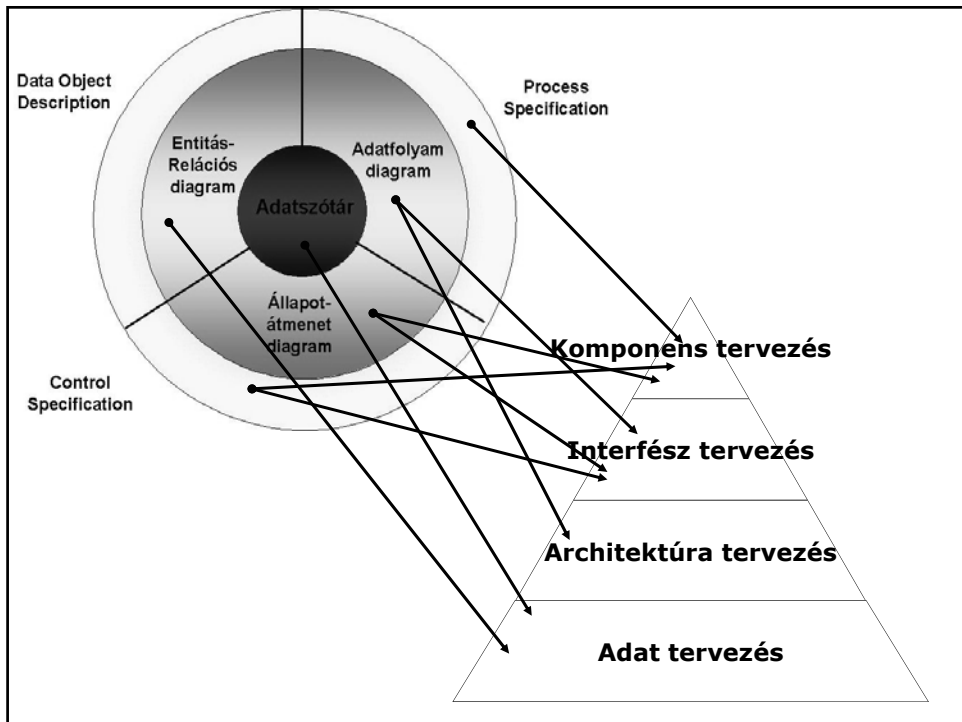
Tervezés fejlődése

- '70-es évek eleje:
 - modularitás, top-down tervezés
- '70-es évek közepe:
 - strukturált programozás, módszertanok
- '70-es évek vége:
 - adatfolyam diagram, adat modellek használata a tervezésben
- '80-as évek:
 - funkció → adat szerkezet
- '90-es évek:
 - objektum orientált szoftver fejlesztés
- '90-es évek vége:
 - szoftver architektúra, minták (pattern) használata



Strukturált tervezés fázisai

- Adat tervezés
 - adat könyvtár, ER → adat struktúrák
- Architektúra tervezés
 - strukturális felépítése a rendszernek
 - → minták, specifikáció, részek...
- Interfész tervezés
 - belső és külső kommunikáció
- Komponens tervezés
 - architektúra terv elemei → implementálható program elemek, procedúrák, függvények stb.



Strukturált tervezés menete

- tipikusan iteratív folyamat
 - absztrakt reprezentáció → konkrét reprezentáció
- tervezési modell
 - teljes reprezentációja a szoftvernek
 - különböző nézetek:
 - adat, funkció, viselkedés



Strukturált tervezés és a minőség

- Tervezés minősége alapvetően befolyásolja a végtermék minőségét
- terv célja:
 - felhasználói követelmények → szoftver rendszer
 - Általános szempontok a terv minősítésére (McGlaughlin):
 - Az összes követelményt lefedi?
 - Érthető, egyértelmű? (fejlesztők, karbantartók)
 - Tartalmazza a szoftvert minden szempontból történő leírását: adat, funkció, viselkedés?
 - minőségi kritériumrumok a tervvel szemben:
 - architektúra tiszta leírása
 - „strukturált”, moduláris leírás
 - elkülöníthető leírása az adat, architektúra, interfész és komponens



Általános tervezési elvek



Általános tervezési elvek I.

- Ne legyen „csőlátású” a tervező
 - alternatívák felállítása
- Az analízis modell és a tervezési modell összekapcsolása
 - melyik tervezési döntés (modell) melyik analízis modell elem alapján jött létre
- „Ne találjuk fel a kereket megint!”
 - tervezési mintákat próbáljunk használni



Általános tervezési elvek II.

- A megoldás „szerkezete” lehetőleg tükrözze a megoldott probléma szerkezetét
- Egységes terv
 - ~egy embertől származna
- Változás tűrő, stukturált
 - új rész integrálható
- Robusztus
 - hiba, túlterhelés stb. esetén lassan csökken a rendszer funkcionalitása



Általános tervezési elvek III.

- a terv absztrakciós szintje magasabb, mint a kódé, és részletes eléggé, hogy ne kelljen lényeges döntést hozni kódoláskor
- minőségi követelményeket figyelembe kell venni és mérni a tervezés folyamatában
 - nem szabad „elhalasztani” a minőségi követelmények megvalósítását
- ellenőrizni kell a tervet a szemantikus hibák kijavítása érdekében (review)
 - ellentmondások, redundancia...
 - nem csak szintaktikusan kell ellenőrizni

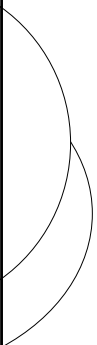


Tervezési módszerek



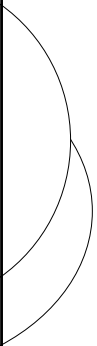
Tervezési módszerekről általában

- Módszerek:
 - segítség a döntéseknél, de...
- Kérdések tervezéskor:
 - Mi alapján lehet a szoftvert részekre osztani?
 - Hogyan legyenek a funkciók, ill. a adatszerkezetek részletei elválasztva a szoftver koncepcionális modelljétől?
 - Van-e általánosan használható mérőszám a tervezés minőségének mérésére?



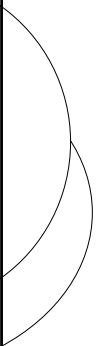
Tervezési módszerek: Absztrakció I.

- Absztrakció
 - tervezés során a terv absztrakciós szintje csökken
 - absztrakt leírás: megoldás (modell) a probléma tér fogalmaival
 - kevésbé absztrakt: keverednek a probléma tér fogalmai az implementációs tér fogalmaival
 - végső terv: közvetlenül megvalósítható megoldás leírása



Tervezési módszerek: Absztrakció II.

- Absztrakció típusai
 - működés leírás (procedural) absztrakció
 - függvény név → függvény utasítások definiálása
 - pl.: felhasználó regisztrálja magát...
 - adat absztrakció
 - adat szerkezet név → adat szerkezet definíció
 - pl.: jelentkező
 - vezérlés absztrakció
 - szinkronizáció események között
 - pl.: párhuzamos (egyidejű) két esemény → semafor



Tervezési módszerek: Pontosítás, részletezés

- refinement - pontosítás
- hierarchikus, top-down tervezési stratégia
 - probléma dekompozíciója, részekre bontása
 - tervezési döntések meghozatalának sorrendje természetes módon adódik, „strukturált”
- elsősorban működés leírás kidolgozására hatékony
- lépésenkénti finomítás, részletezése a megoldás leírásának

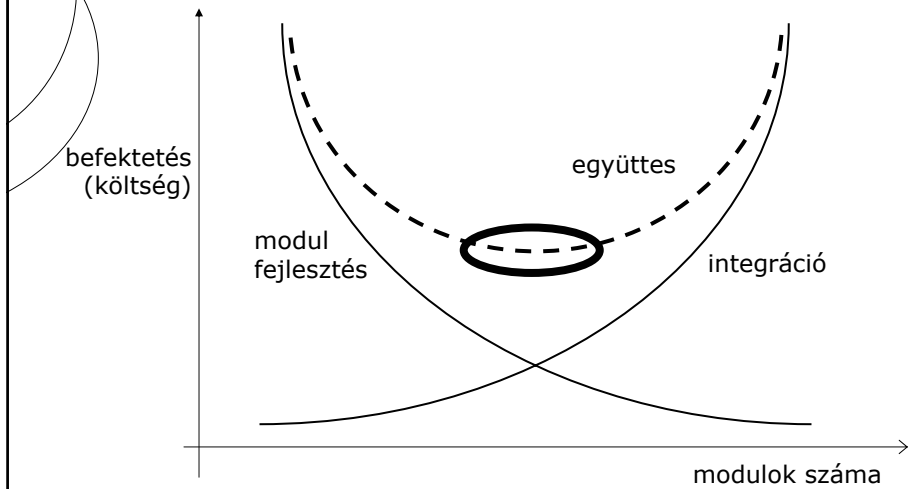
Tervezési módszerek: Modulokba szervezés

- komponensek névvel és elhatárolható funkcionalitással rendelkező részek, melyek önálló megvalósítással rendelkeznek
- kezelhető legyen a rendszer „józan ésszel”

Modulok méretének meghatározása

- Probléma: p_1, p_2
- Komplexitás: $K(p_1), K(p_2)$
- Befektetés (költség): $B(p_1), B(p_2)$
- $K(p_1) > K(p_2) \rightarrow B(p_1) > B(p_2)$
- **$K(p_1 + p_2) > K(p_1) + K(p_2) \rightarrow$
 $B(p_1 + p_2) > B(p_1) + B(p_2)$**
- \rightarrow minél kisebb problémákra vágom, annál könnyebben oldom meg...
- de integrálni is kell \rightarrow optimum!!

Modulok mérete – fejlesztés költsége



Szoftver architektúra

Szoftver architektúra

- rendszer általános felépítése, struktúrája
- komponensek hierarchiája
 - komponensek mérete nem meghatározott!!
- komponensek együttműködésének módja
- *A szoftver architektúra leírása koncepcionális leírás*

Szoftver architektúra leírás részei

- **strukturális modell:** rendszer felépítése
- **„framework” modell:**
 - Absztrakt: milyen általános működési megoldásokat, „tervezési mintákat” használunk a rendszerünkben.
- **dinamikus modell:**
 - Rendszer viselkedésének leírása:
 - a rendszer hogyan változik külső hatásokra.
- **folymat modell:**
 - Milyen üzleti vagy technológiai folyamatot támogat (valósít meg) a rendszer.
- **funkcionális modell:**
 - A rendszer által megvalósított funkciók hierarchiája.
- architektúra leírása:
 - különböző „Arhitectural Description Language” (Architektúra Leíró Nyelvek) segítségével



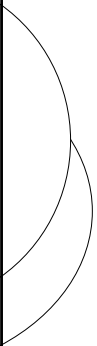
Strukturális modell

- Két szempontból beszélhetünk a szoftver rendszer struktúrájáról:
 - **Vezérlési elemek struktúrája, hierarchiája**
 - **Adat elemek struktúrája, hierarchiája**



Struktúra: **Vezérlési hierarchia**

- modulok aktiválási sorrendje, alternatívái
 - fa szerkezetű ábrázolás
 - Jackson diagram
- meghatározható a vezérlés bonyolultsága
- definiálható a modulok láthatósága, kapcsolata



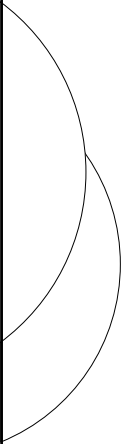
Architektúra meghatározása vezérlés alapján

- **Horizontális bontás**
 - bontás egy-egy külső funkció alapján
 - könnyű tesztelni, kevés mellékhatás változtatáskor, bővíthető
- **Vertikális bontás**
 - vezérlő és munkavégző modulok
 - változás általában a munkavégző modulokban → karbantarthatóbb (kevesebb mellékhatás)



Struktúra: **Adat elemek hierarchiája, struktúrája**

- alap adattípusok
- adatszerkezetek:
 - vektorok
 - több dimenziós tömbök
 - láncolt listák
- hierarchikus adatszerkezetek



Általános elvek hatékony struktúra, ill. hierarchia tervezéséhez



Információ rejtés

- modularitás gyakorlati haszna
- modulok „önálló egységek”
- önállóan lehessen őket tervezni megvalósítani
- belső működés, szerkezet rejtett a külvilág elől
 - adatszerkezetek
 - vezérlés



Modulok tervezése

- Funkcionális függetlenség:
 - kohézió (összetartozás)
 - csatolás
- **Kohézió**
 - együttműködés mértéke egy vagy több feladat (funkció) megvalósításában
 - esetleges, logikai, állandó, procedurális...
- **Csatolás**
 - interfész bonyolultsága alapján
 - hívási paramétereken keresztül
 - vezérlő adatszerkezeteken keresztül
 - globális adatszerkezeteken keresztül
 - környezeti elemeken (eszközökön) keresztül



Szoftver architektúra tervezés



Szoftver architektúra tervezés

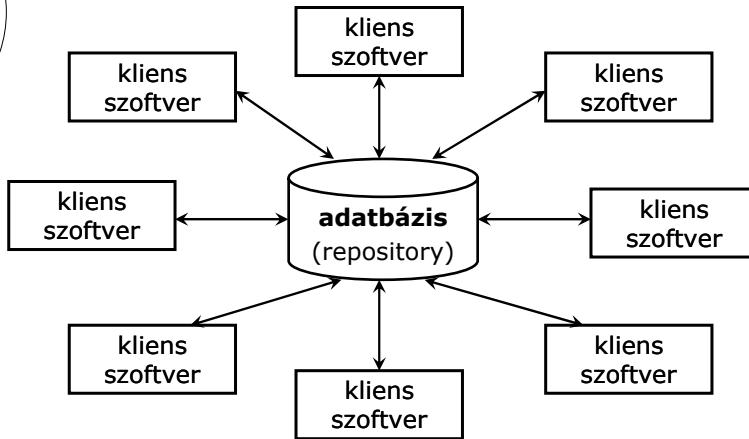
- rendszer általános működésére, felépítésére vonatkozó legfontosabb (korai) döntések
- követelmények megvalósítása
- alternatívák számbavétele
- megvalósítás rizikójának csökkentése
- érthető méretű leírás: kommunikáció



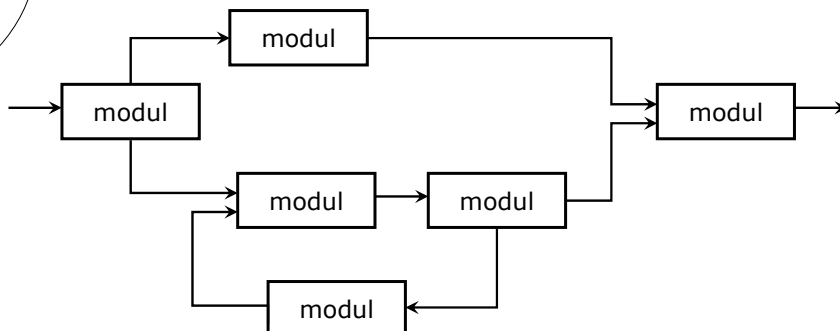
Tipikus architektúrák, „stílusok”

- Adat központú architektúra
 - Adat tároló központ + kliensek
- Adatfolyam (data flow)
 - pipe, batch ...
- „Call and return” architektúra
 - program – alprogram
 - remote procedure call (kliens-szerver)
- Objektum orientált architektúra
- Rétegszerkezet

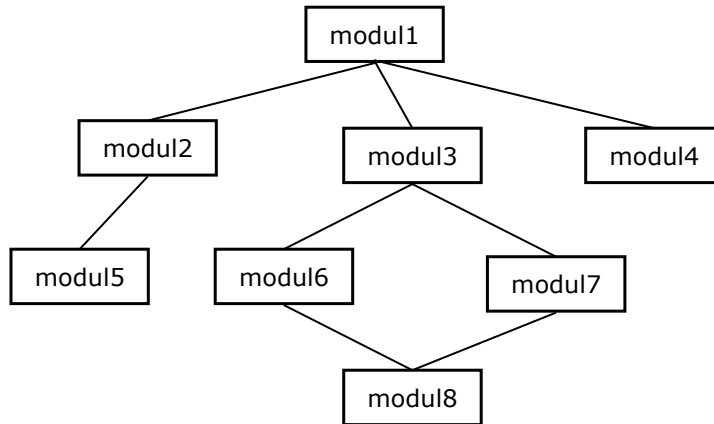
Adat központú architektúra



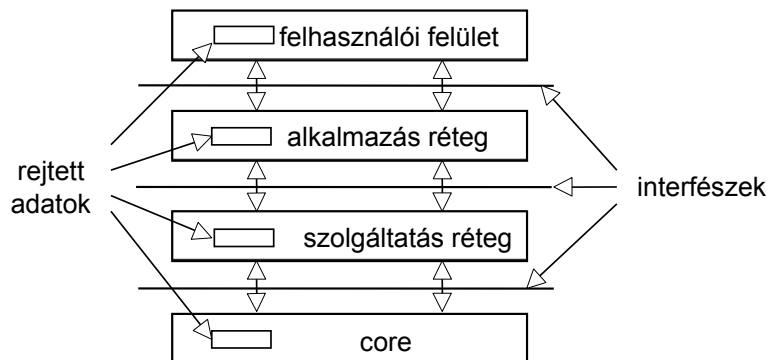
Adatfolyam (data flow) típusú feldolgozás



„Call and return” architektúra




Rétegszerkezet





Interfész tervezés



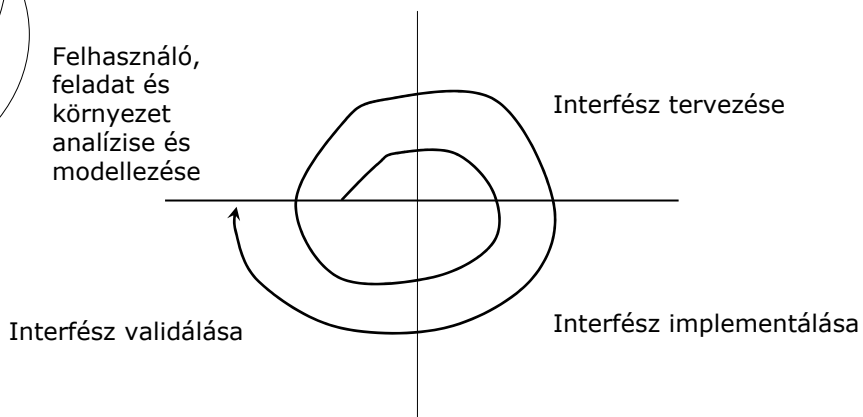
Interfész tervezés

- A rendszer végső használhatósága szempontjából fontos
- Forgatókönyvek készítése (pl. use case-ek alapján)
 - Interfész objektumok és műveletek definiálása
- Szempontok a tervezéshez
 - Felhasználó „modellje”: kezdő, haladó, tapasztalt

Interfész tervezés folyamata

- Négy lépés:
 - Felhasználó, feladat és környezet analízise és modellezése
 - Interfész tervezése
 - Interfész implementálása
 - Interfész validálása
- *Tipikusan iteratív folyamat!!*
- Általános módszer:
 - prototípus készítés

Interfész tervezés iteratív folyamata



Komponens tervezés

Komponens tervezés



○ Eredmény:

- részletes leírása a komponenseknek
- a megvalósítás részletei
- grafikus v. szöveges leírása a megvalósítandó kódnak



Strukturált tervezés fázisai (volt)

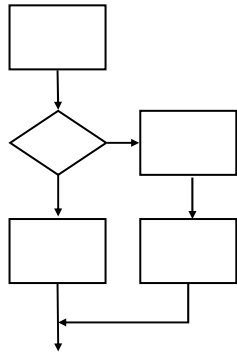
- **Architektúra kialakítása:**
 - **Architektúra tervezés**
 - strukturális felépítése a rendszernek
 - → minták, specifikáció, részek...
 - **Adat tervezés**
 - adat könyvtár, ER → adat struktúrák
- **Részletes tervezés:**
 - **Interfész tervezés**
 - belső és külső kommunikáció
 - **Komponens tervezés**
 - architektúra terv elemei → implementálható program elemek, procedúrák, függvények stb.



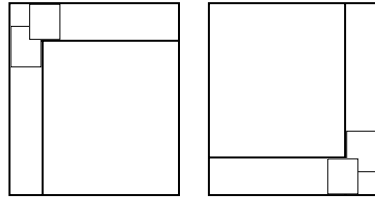
Komponens tervezés módszerei

- **procedural design**
 - döntések a részletekig
- **használható modellek**
 - folyamat vezérlési gráf (control flow graph)
 - vezérlési szerkezet
 - dobozos jelölés (box notation)
 - vezérlési szerkezetek
 - döntési táblák
 - szabályok rögzítése
 - feltételek (pl. bemenetek) → hatások (műveletek)
 - **Program Design Language**
 - „szimbolikus” program kód

Komponens tervezési modellek



folyamat vezérlési gráf



box notation