



# Software Engineering

---



## Software Engineering

---

### **i 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...



## Modellezés

---

- | **Általános mérnöki tevékenység**
- | **Modellezés:**
  - | A valóság adott szempont szerinti tömör leírása.
  - | Az elemzési szempont alapján számunkra fontos információt kiemeljük, általánosítunk.
  - | Segít a valóság (működésének, felépítésének, stb.) megértésében, elemzésében.



## 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 előállítása érdekében tett műveletek (pl. modellezés, kódolás, tervezés stb.)
- | **Fejlesztési termékek:**
  - | fejlesztési lépések eredményei
  - | pl. modellek, kód, dokumentáció stb.

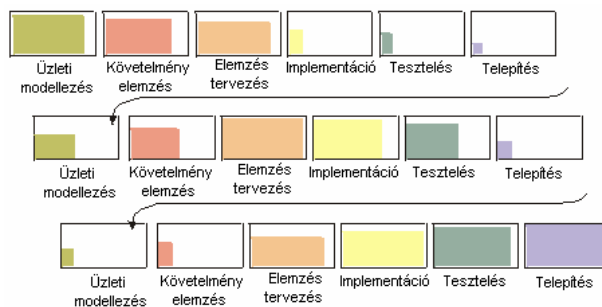
## Fejlesztés lépéseinek sorrendje

### i Vízesés modell szerint



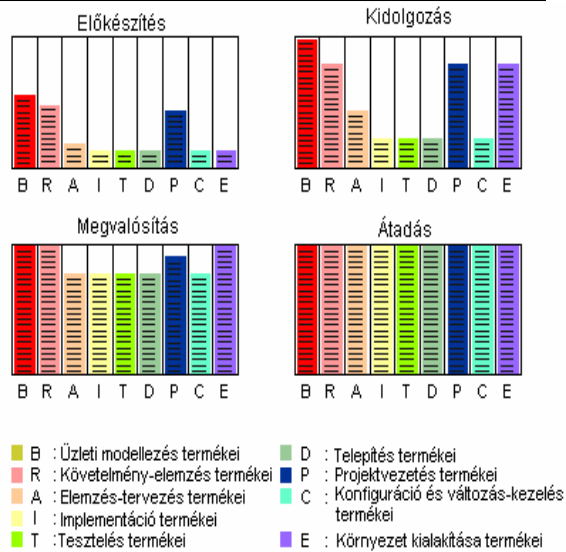
## Fejlesztés lépéseinek sorrendje

### i 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.



## 1. System Engineering

## System Engineering

| SYSTEM ENGINEERING                 |                                     |                               |                                       |
|------------------------------------|-------------------------------------|-------------------------------|---------------------------------------|
| <i>Product Engineering</i>         | <i>Business Process Engineering</i> |                               |                                       |
| Requirement Engineering (Business) | Information System Planning         |                               | Üzleti modellezés (Termék modellezés) |
|                                    | Business Area Analysis              |                               |                                       |
| Requirement Engineering (Software) | Business System Design              | } <b>Software Engineering</b> | Követelmény (elemzés)                 |
| System Component Engineering       |                                     |                               | Construction & Integration            |
|                                    | Tervezés                            |                               |                                       |
|                                    | Implementáció                       |                               |                                       |
|                                    | Tesztelés                           |                               |                                       |
|                                    |                                     |                               | Üzembe helyezés                       |

## System Engineering (SysEng)

- i Általános értelemben vett rendszer fejlesztés
- i System Engineering lehet
  - i Business Process Engineering  
ha vállalat (működésének) (át)szervezésével foglalkozunk  
Pl.: számlázási rendszer,...
  - i 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.**
- | **Részei:**
  - | szoftver
  - | hardver
  - | emberek (felhasználók, operátorok)
  - | dokumentáció (rendszer leírása)
  - | eljárások (a rendszer használatának lépései )



### 1.1. Business Process Engineering (BPE)

---

**Üzleti folyamatok leírása, elemzése, tervezése**



## Business Process Engineering

| SYSTEM ENGINEERING                 |                              |                       |                                       |
|------------------------------------|------------------------------|-----------------------|---------------------------------------|
| Product Engineering                | Business Process Engineering |                       |                                       |
| Requirement Engineering (Business) | Information System Planning  | Software Engineering  | Üzleti modellezés (Termék modellezés) |
|                                    | Business Area Analysis       |                       |                                       |
| Requirement Engineering (Software) | Business System Design       |                       |                                       |
| System Component Engineering       |                              |                       | Construction & Integration            |
|                                    |                              |                       |                                       |
|                                    |                              |                       |                                       |
|                                    |                              | Követelmény (elemzés) |                                       |
|                                    |                              | Elemzés               |                                       |
|                                    |                              | Tervezés              |                                       |
|                                    |                              | Implementáció         |                                       |
|                                    |                              | Tesztelés             |                                       |
|                                    |                              | Üzembe helyezés       |                                       |

## Business Process Engineering (BPE)

### i Célja:

- Olyan felépítés (architektúrák) definiálása az üzleti tevékenység számára, mely lehetővé teszi az információ hatékony felhasználását és ezáltal az üzlet hatékonyabb működését.

### i 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**
- | **Szemponatok:**
  - | **adat felépítés (architektúra)**
    - | üzleti folyamatok által kezelt (feldolgozott) adatok, köztük levő kapcsolatok
  - | **applikációs felépítés (architektúra)**
    - | adatok feldolgozásának folyamata
    - | adatokat feldolgozó elemek működése
      - | szoftverek, emberek
  - | **technológiai felépítés (architektúra)**
    - | szoftver, hardver elemek, melyek az adatok tárolását és az applikációk működését támogatják



## Adat felépítés (architektúra)

---

- | **Példa**
  - | **Adatok, amelyeket a szervezet működtetéséhez kezelni kell.**
  - | **Ügyfél**
    - | név
    - | cím, kontakt információ
    - | korábbi vásárlások
    - | utolsó vásárlás
  - | **Termék**
    - | ...
  - | ...
  - | **Kapcsolatok leírása a tárolt adatok között.**



## Applikációs felépítés (architektúra)

---

### i Példa

- | Bejövő számlákat a pénzügyes feldolgozza, a törzsadatokat beviszi a számlanyilvántartó rendszerbe, a számlákat archiválja.
- | A számlaadatok alapján a számlanyilvántartó rendszer banki utalást állít össze. Az utalást az üzletvezető engedélyezi és indítja.
- | ...



## Technológiai felépítés (architektúra)

---

### i Példa

- | A vállalatnál egy ProLiant ML350-es szerveren (tároló: 162 GB, RAID 2,...) fut egy testreszabott ERP rendszer (Infosys), a banki rendszerrel közvetlen VPN kapcsolat van, fizikai közeg: SDSL, 2 Gbit/s, ...

## BPE lépései (szintjei) I.

### i Information Strategy Planning (ISP)

- | Információs (információ kezelési) stratégia tervezés
- | üzleti területek (domain-ek) meghatározása (pl. gyártás, marketing, pénzügyek stb.)

### i 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.

### i Business System Design (BSD)

- | Üzleti rendszerek tervezése
- | à Software engineering
- | Követelmények részletes és pontos meghatározása az alkalmazandó (információrendszer) számítógépes rendszer számára.

### i Construction and Integration (C&I)

- | Megvalósítás és integrálás
- | Alkalmazás megvalósítása, integrálása az adott környezetbe.

## 1.2. Product Engineering

### Product Engineering

| SYSTEM ENGINEERING                 |                                     |                             |                                       |
|------------------------------------|-------------------------------------|-----------------------------|---------------------------------------|
| <b>Product Engineering</b>         | <b>Business Process Engineering</b> |                             |                                       |
| Requirement Engineering (Business) | Information System Planning         |                             | Üzleti modellezés (Termék modellezés) |
|                                    | Business Area Analysis              |                             |                                       |
| Requirement Engineering (Software) | Business System Design              | <b>Software Engineering</b> | Követelmény (elemzés)                 |
|                                    |                                     |                             | Elemzés                               |
|                                    |                                     |                             | Tervezés                              |
|                                    |                                     |                             | Implementáció                         |
|                                    |                                     |                             | Tesztelés                             |
| System Component Engineering       | Construction & Integration          |                             | Üzembe helyezés                       |



## Product Engineering

---

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



## Product Engineering lépései

---

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



## 1.3. Requirement Engineering

---

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



## Requirement Engineering (RE)

---

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



## Requirement Engineering (RE)

---

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



## Requirement Engineering lépései

---

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





## Követelmények kiderítése, összegyűjtése

---

### i Feladatok:

- | **Megoldandó probléma határának meghatározása (scope)**
- | **Követelmények értelmezése (understanding)**
  - | értelmezni, milyen lehetőségei lesznek a rendszernek,
  - | triviális és félreérthető követelmények elhagyása,
  - | ellenőrizhető követelmények definiálása
- | **Változó követelmények kezelése (volatility)**
  - | ha a követelmények változnak ezt kezelni kell



## Követelmények elemzése és egyeztetése

---

- i kategorizálás, követelmény típusok definiálása
  - | funkcionális
  - | kezelt adatokra vonatkozó
  - | viselkedésre vonatkozó
  - | felhasználói felületre vonatkozó
  - | kimenetekre vonatkozó
  - | ...
- i követelmények közötti összefüggések meghatározása:
  - | konzisztencia,
  - | prioritások, ...
- i különböző forrásból származó (stakeholder) követelmények összehangolása
- i követelményekhez kapcsolódó (megvalósításhoz) kockázatok tisztázása



## Követelmény specifikáció készítése

---

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



## Rendszer modellezés

---

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



## Követelmények validálása

---

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



## Követelmények változásának kezelése, követése

---

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



## 2. Elemzés (Követelmény elemzés)

---

**Software requirement engineering**



### Elemzés

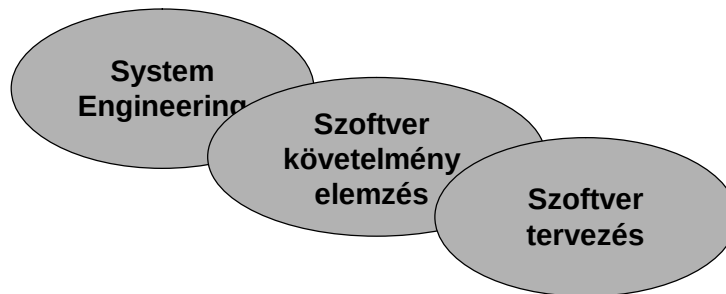
---

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

## Követelmény elemzés szerepe a szoftver fejlesztésben

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

## Elemzés kapcsolata a szoftver rendszer fejlesztés lépéseivel





## Követelmény elemzés általános lépései

---

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



## 2.1. Követelmények kiderítése, összegyűjtése szoftver fejlesztéskor

---



## Követelmények kiderítése, összegyűjtése

---

### i **interaktív rész**

- | beszélgetés, interjú, workshop,...

### i **cél:**

- | kideríteni a „részvényeseket” – akik húznak hasznot a rendszerből (felhasználók, megrendelő, stb.)
- | igények: funkció, kimenet, teljesítmény, ...
- | megfelelő embert kérdeztünk, van-e aki jobban tudja definiálni



## Technikák követelmények gyűjtésére

---

### i **Facilitated Application Specification Technique (FAST)**

- | Felhasználókat és sw fejlesztőket „egy oldalra gyűjtöm”
- | formalizált lépések a követelmények definiálására a két oldal összehangolásával

### i **Quality Function Deployment**

- | módszer arra, hogy a felhasználói követelményeket a megvalósítandó rendszerre vonatkozó követelményekké fordítsuk át



## Facilitated Application Specification Technique (FAST)

---

### **Megkönnyített Alkalmazás Specifikáció Készítő Módszer**

- i Probléma:**
  - | Fejlesztők és a megrendelők nehezen értik meg egymást.
- i Megoldás:**
  - | megbeszélés szervezése
  - | szabályok felállítása a résztvevők számára
  - | napirend készítése
  - | vezető (facilitator) kiválasztása
  - | követelmény definiáló módszer kiválasztása
    - | tábla, elektronikus, formalizmus ...
  - | Cél, hogy közösen definiáljanak követelményeket, elemezzenek különböző megoldási lehetőségeket



## Quality Function Deployment

---

- i Funkciók kategorizálása**
- i A követelmények három kategóriába sorolása:**
  - | felhasználó által közvetlenül definiált követelmények
  - | elvárt (implicit) követelmények, melyeket nem feltétlenül definiálja a felhasználó, mert természetesnek veszi
  - | „izgalmas” követelmények
    - | a közvetlenül definiált követelmények feletti követelmények, ami igazán elégedetté teszi a felhasználót





## Quality Function Deployment

---

- i **Funkciókra vonatkozó követelmények kategorizálása**
- i **Feldolgozott információkra vonatkozó követelmények kategorizálása**
  - | adatok és események, melyek a rendszer felhasznál, ill. előállít
- i **A rendszer által végrehajtott feladatokra vonatkozó követelmények kategorizálása**
  - | viselkedéssel kapcsolatos követelmények
- i **Érték elemzés**
  - | felsorolt és kategorizált követelmények relatív összehasonlítása



## Technika funkcionális követelmények összegyűjtésére: Use case-ek

---

- i **aktor**
  - | rendszerrel kapcsolatba kerülő felhasználó v. rendszer
  - | valamit „akar” a rendszertől: input, output
  - | szerepkört definiál!
- i **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)

---

**i 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

---

- i **sw alkalmazás ~ adat feldolgozás, konverzió**
  - | bemenet à kimenet
  - | adat + vezérlő információ
- i **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)

---

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



## Részekre bontás (gy)

---

- | **komplex probléma à 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

---

- i **alapelvek, általános szabályok a specifikáció készítésre vonatkozóan:**
  - | funkció és az implementáció szétválasztása
  - | rendszer viselkedésének (funkció, adatkonverzió) leírása külső események hatására
  - | ...
- i **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 (β „információs domain”)
    - ┆ rendszer által kezelt információk, adatok, köztük levő kapcsolatok leírása
  - ┆ Funkció leírása (β „functional domain”)
    - ┆ rendszer által megvalósítandó műveletek a kezelt adatokon
  - ┆ Viselkedés leírása (β „behavioral domain”)
    - ┆ a végrehajtott funkciók időbeni végrehajtási sorrendje
  - ┆ Rendszer (v. funkció) validációjának lehetséges módja (fontos a teszteléshez!!)



## 2.4. Review

---



## Review

---

### i SRS

- | teljes, konzisztens, pontos ???
- | általában „manuálisan”
  - i nehéz!!
  - i prototípus à működtetés
  - i modell à szimuláció
- | megrendelővel jóváhagyatni
  - i a szerződés alapja J



## 3. Elemzés módszerei

---

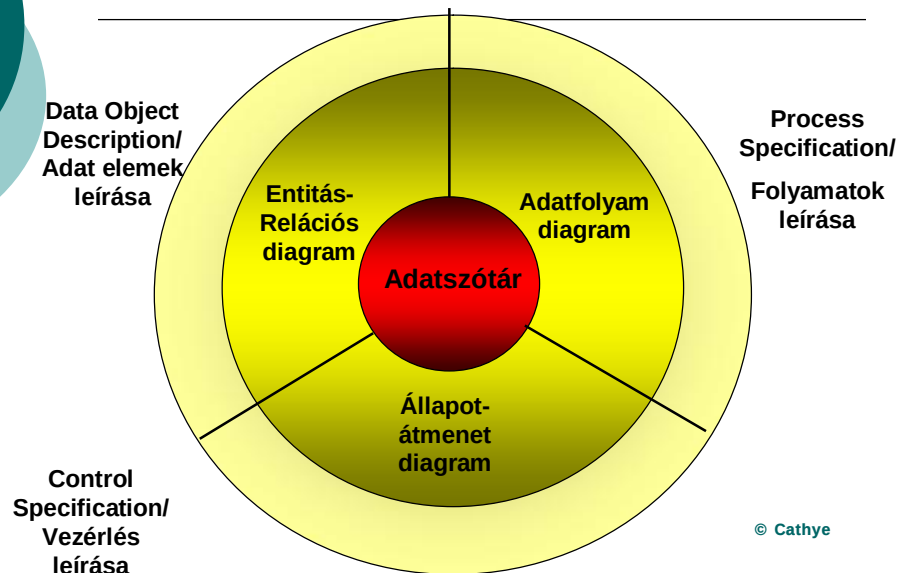


## Elemzés

### i Elemzés (teljes fejlesztés) megvalósítási alternatívái

- strukturált à procedurális program fejlesztés
- objektum orientált à objektum orientált rendszer fejlesztése

## Strukturált elemzés során készített modell szerkezete





## Strukturált elemzés eszközei

---

- i **Adat könyvtár (dictionary)**
  - | Rendszer által kezelt adatok leírása
- i **Entitás relációs diagram (entity relationship diagram)**
  - | Adat objektumok és az adatelemek egymáshoz való viszonyának leírása
- i **Adat folyam diagram (dataflow)**
  - | Adat transzformáció és adat mozgatás, valamint az adat manipuláló funkciók leírása
- i **Állapot átmeneti diagram (state transition diagram)**
  - | Vezérlés leírása



## Strukturált elemzés – GYAKORLATON!

---

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



## 4. Tervezés

---



## Tervezés

---

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



## Tervezés fejlődése

---

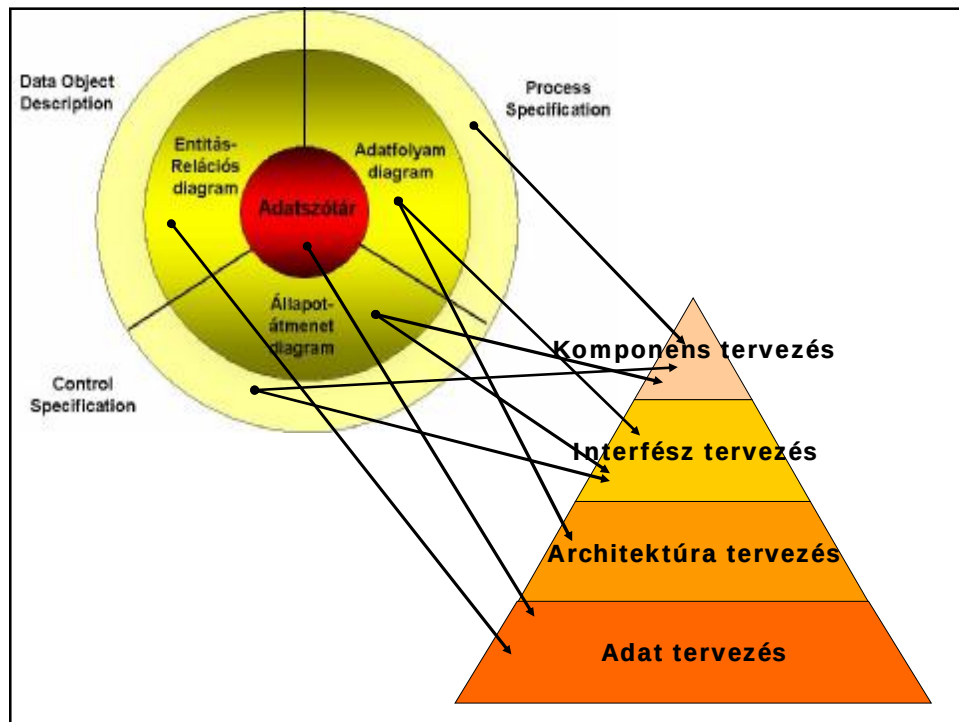
- i '70-es évek eleje:
  - | modularitás, top-down tervezés
- i '70-es évek közepe:
  - | strukturált programozás, módszertanok
- i '70-es évek vége:
  - | adatfolyam diagram, adat modellek használata a tervezésben
- i '80-as évek:
  - | funkció à adat szerkezet
- i '90-es évek:
  - | objektum orientált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



## Strukturált tervezés fázisai

---

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



## Strukturált tervezés menete

### i tipikusan iteratív folyamat

- ▮ absztrakt reprezentáció à konkrét reprezentáció

### i tervezési modell

- ▮ teljes reprezentációja a szoftvernek
- ▮ különböző nézetek:
  - i 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



## 4.3. Szoftver architektúra

---



## Szoftver architektúra

---

- i rendszer általános felépítése, struktúrája
- i a rendszert felépítő komponensek egymáshoz való viszonya (hierarchiája, szervezése)
  - i az architektúra leírásban szereplő komponensek mérete nem meghatározott
- i komponensek együttműködésének módja
  - i komponensek kapcsolatai
- i ***A szoftver architektúra leírás koncepcionális leírás***



## Szoftver architektúra leírás részei, nézetei

---

- i **strukturális modell / nézet: rendszer felépítése**
- i **„framework” modell / nézet:**
  - i Absztrakt: milyen általános működési megoldásokat, „tervezési mintákat” használunk a rendszerünkben.
- i **dinamikus modell / nézet:**
  - i Rendszer időbeli viselkedésének leírása:
    - i a rendszer hogyan változik külső hatásokra.
- i **funkcionális modell / nézet:**
  - i A rendszer által megvalósított funkciók (bemeneti, ill. kimeneti adatok közötti kapcsolat).
- i **folyamat modell / nézet:**
  - i A rendszer által támogatott, ill. megvalósított üzleti vagy technológiai folyamat leírása.
- i **architektúra leírása:**
  - i 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

---

### i **Horizontális bontás**

- | bontás egy-egy külső funkció alapján
- | könnyű tesztelni, kevés mellékhatás változtatáskor, bővíthető

### i **Vertikális bontás**

- | vezérlő és munkavégző modulok
- | változás általában a munkavégző modulokban → karbantarthatóbb (kevesebb mellékhatás)



## Struktúra: *Adat elemek hierarchiája, struktúrája*

---

### i **alap adattípusok**

### i **adatszerkezetek:**

- | vektorok
- | több dimenziós tömbök
- | láncolt listák

### i **hierarchikus adatszerkezetek**



## 4.4. Általános elvek hatékony struktúra, ill. hierarchia tervezéséhez

---



### Információ rejtés

---

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



## Modulok tervezése

---

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



## 4.5. Szoftver architektúra tervezés

---



## Szoftver architektúra tervezés

---

- i **rendszer általános működésére, felépítésére vonatkozó legfontosabb (korai) döntések**
- i **követelményeket kielégítő rendszer megvalósítása**
- i **alternatívák számbavétele**
- i **cél a megvalósítás rizikójának csökkentése**
- i **érthető méretű leírás: kommunikáció**



## Szoftver architektúra tervezés fontossága

---

- i **Azonosítani a fontos komponenseket, áttekintést adni a rendszerről**
- i **A fejlesztés korai fázisában meghatározni a rendszer fő jellemzőit**
- i **Robusztus rendszer kialakítása**
  - i **robusztus: jól tűri a változásokat, rugalmasan fejleszthető a meglévő rendszer változtatása nélkül**
- i **Újrafelhasználható rendszer komponensek készítése**



## Tipikus architektúrák, elterjedt architektúra modellek

---



## Tipikus architektúrák, elterjedt architektúra modellek

---

- **Adat központú architektúra**
- **Adatfolyam (data flow)**
- **„Call and return” architektúra**
- **Kliens-szerver architektúra**
- **Rétegszerkezet**



## Tipikus architektúrák, elterjedt architektúra modellek

---

### **| Rétegszerkezet**

- | felhasználói felület**
- | alkalmazás réteg**
  - | az adott alkalmazásra jellemző működés megvalósítása**
- | szolgáltatás réteg**
  - | alapfunkciók, melyek ritkán változnak**
  - | üzleti logika, adatbázis kezelés logikája**
- | mag/core**
  - | operációs rendszer, adattárolás, stb.**

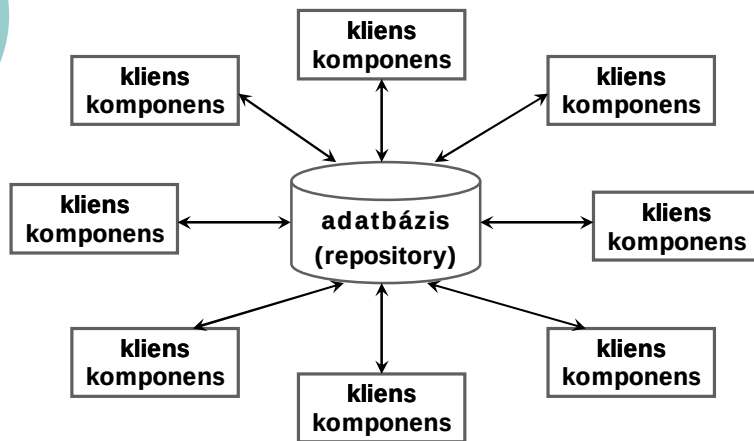


## Adat központú architektúra

---

- | Adat tároló központban megosztott adatok + kliensek, akik az adatokat felhasználják**

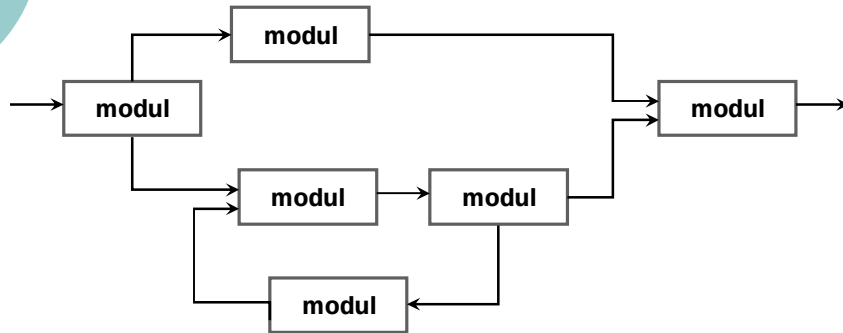
## Adat központú architektúra



## Adatfolyam (data flow)

- i lehet pipe vagy batch jellegű
- i komponensek, amelyek az adatokat sorban feldolgozzák

## Adatfolyam (data flow) típusú feldolgozás

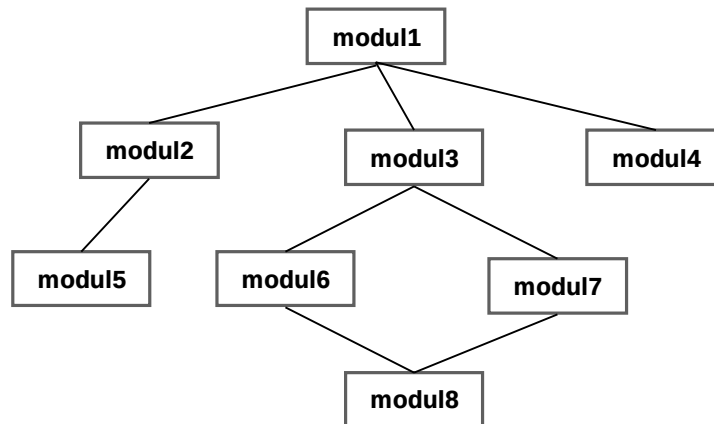


## „Call and return” architektúra

- i program – alprogram
- i remote procedure call (kliens-szerver)



## „Call and return” architektúra

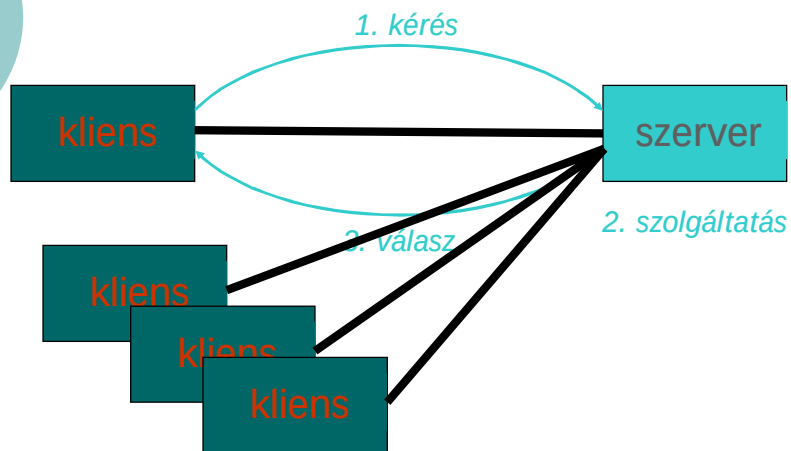


## Kliens-szerver architektúra

### i Kliens-szerver működés:

- Modulok:  
szolgáltatások nyújtása és használata.
- Szolgáltatások igénybevétele  
kommunikációs csatornán keresztül.

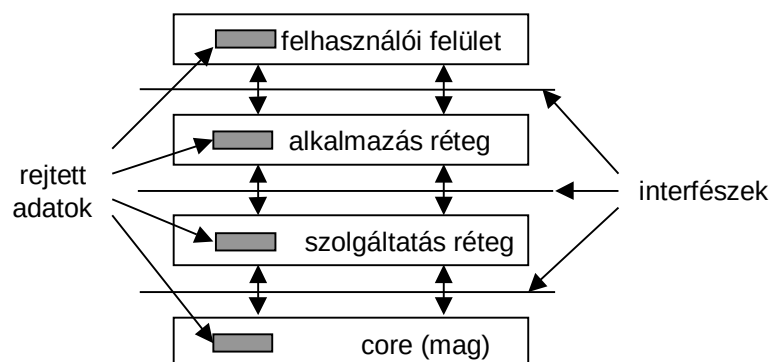
## Kliens-szerver működés



## Rétegszerkezet

- i **felhasználói felület**
- i **alkalmazás réteg**
  - | az adott alkalmazásra jellemző működés megvalósítása
- i **szolgáltatás réteg**
  - | alapfunkciók, melyek ritkán változnak
  - | üzleti logika, adatbázis kezelés logikája
- i **mag/core**
  - | operációs rendszer, adattárolás, stb.

## Rétegszerkezet



## 4.6. Interfész tervezés



## Interfész tervezés

---

### i **Interfész**

#### | **felhasználói interfész**

##### | **kapcsolat a külvilággal**

###### | **felhasználók**

###### | **egyéb szoftver rendszerek**

#### | **komponensek közötti interfész**

##### | **modulok egymás közötti kommunikációja**

##### | **komponens tervezéssel összehangolt**



## Felhasználói interfész tervezés

---

### i **A rendszer végső használhatósága szempontjából fontos**

### i **Forgatókönyvek készítése (pl. use case-ek alapján)**

#### | **Interfészt megvalósító elemek és általuk végrehajtott műveletek definiálása**

### i **Szemponatok a tervezéshez**

#### | **Felhasználó „modellje”: kezdő, haladó, tapasztalt**

## Felhasználói interfész tervezés folyamata

### i Négy lépés:

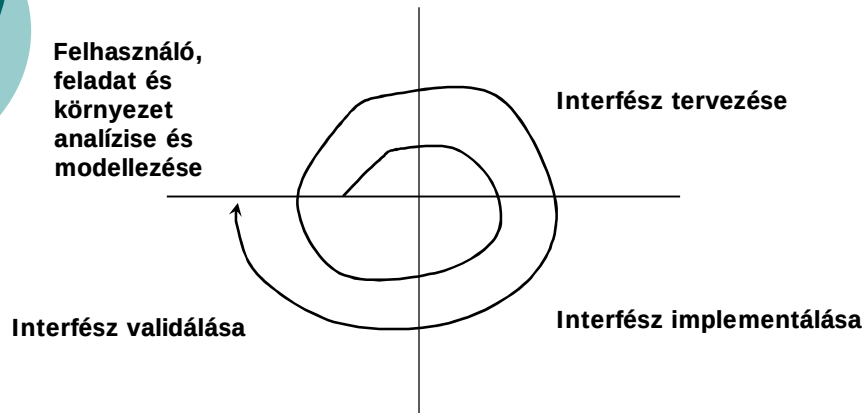
- | Felhasználó, feladat és környezet analízise és modellezése
- | Interfész tervezése
- | Interfész implementálása
- | Interfész validálása

### i *Tipikusan iteratív folyamat!!*

### i Általános módszer:

- | prototípus készítés

## Interfész tervezés iteratív folyamata



## 4.7. Komponens tervezés

### Komponens tervezés



#### i Eredmény:

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

## Strukturált tervezés fázisai (volt)

### i **Architektúra kialakítása:**

- i **Adat tervezés**
  - i adat könyvtár, ER à adat struktúrák
- i **Architektúra tervezés**
  - i strukturális felépítése a rendszernek
  - i à minták, specifikáció, részek...

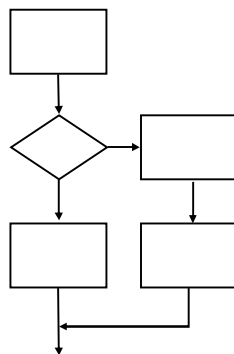
### i **Részletes tervezés:**

- i **Interfész tervezés**
  - i belső és külső kommunikáció
- i **Komponens tervezés**
  - i architektúra terv elemei à implementálható program elemek, procedúrák, függvények stb.

## Komponens tervezés módszerei

- i **procedural design**
  - i döntések a részletekig
- i **használató modellek**
  - i folyamat vezérlési gráf (control flow graph)
    - i vezérlési szerkezet
  - i dobozos jelölés (box notation) - struktogram
    - i vezérlési szerkezet hierarcikusan
    - i Nassi Schneider diagram
  - i döntési táblák
    - i szabályok rögzítése
    - i feltételek (pl. bemenetek) à hatások (műveletek)
  - i Program Design Language
    - i „szimbolikus” program kód

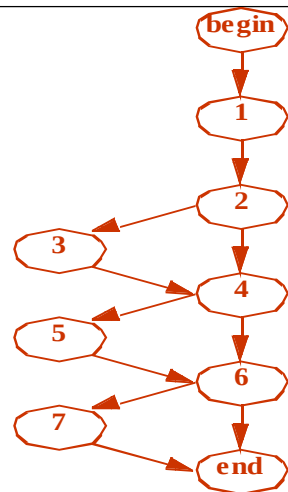
## Komponens tervezési modellek: folyamat vezérlési gráf



folyamat vezérlési gráf

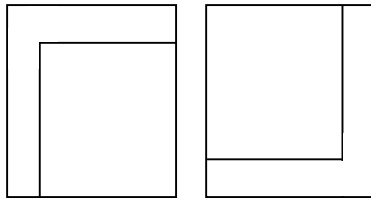
## Control Flow Graph

```
1: mid= (low + high) / 2;  
2: if (x == list(mid)) then  
3:     found = 1;  
4: if (x > list(mid)) then  
5:     low = mid+1;  
6: if (x < list(mid)) then  
7:     high = mid;
```





## Komponens tervezési modellek: Nassi Schneider diagram



box notation  
Nassi Schneider diagram



## Jackson ábra

