




Software Engineering Szoftver fejlesztés

Követelmény (kezelés, elemzés, specifikáció)
Elemzés
Tervezés
(Architektúra)



Engineering (Fejlesztés)

- System Engineering
 - Business process engineering
 - üzleti folyamatok tervezése, szervezése
 - modellezzük az üzleti környezetet
 - Product engineering
 - termékek tervezése
 - modellezzük a terméket, annak használatát
- Software Engineering
 - szoftver alkalmazásokat, eszközöket ad a fenti feladatok megoldására
- Modellezés – általános eszköz
 - leendő rendszer specifikációja és terve



Software Engineering Szoftver fejlesztés

Technikai tartalom, lépések



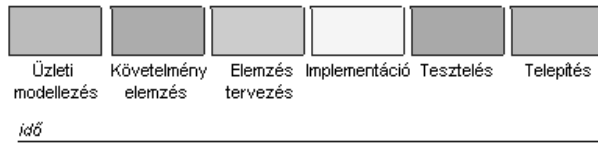
Szoftver Engineering – lépések

- (Üzleti modellezés)
- **Követelmény** (kezelés, elemzés)
- **Elemzés, tervezés**
- **Implementáció**
- **Tesztelés**
- **Telepítés**

- Fejlesztési termékek:
 - fejlesztési lépések eredményei

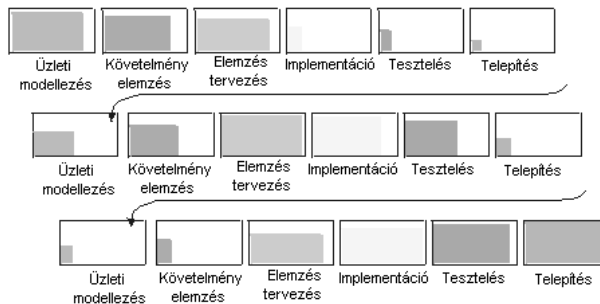
Fejlesztés lépései

o Vizesés modell szerint



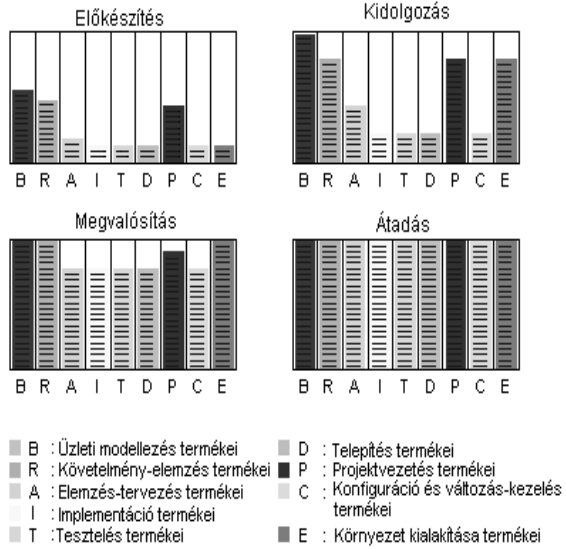
Fejlesztés lépései

o Iteratív fejlesztési modell szerint



Termékek előállítása a fejlesztés során

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



Szoftver Engineering Szoftver fejlesztés

Követelmények

Követelmények

- Követelmények
 - Követelmények összegyűjtése
 - Követelmények elemzése
 - konzisztencia, prioritások...
 - Követelmény specifikáció
 - konzisztens és érthető követelmény definíció (pl. bemenet, kimenet stb.)
 - rendszer modell
 - Követelmények validálása
 - teljesség, ellentmondás-mentesség...

Követelmények – Módszerek

- Követelmények összegyűjtése
 - pl. használati esetek; nem funkcionális...
- Követelmények elemzése
 - information domain
 - adat feldolgozás
 - vezérlés
 - funkció leírás – funkcionális modell
 - viselkedés leírás – viselkedési modell
 - partíciónálás (részekre bontás)
 - prototípus készítés



Szoftver Engineering

Szoftver fejlesztés

Elemzés



Elemzés

- Cél: elemzési modell készítése
- Fejlesztett rendszer részletes és teljes leírása
- első technikai reprezentáció (modell)
- Alternatívák:
 - Strukturált analízis
 - Objektum orientált analízis



Strukturált elemzés

- Elemzési modell célja:
 - felhasználói követelmények rögzítése,
 - a szoftver terv (ill. tervezés) alapjainak megteremtése,
 - követelmények definiálása, melyek alapján verifikálható lesz az elkészített szoftver.



Strukturált analízis módszerei

- 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 mozgatás, valamint az adat manipuláló funkciók leírása
- Állapot átmeneti diagram (state trans.)
 - Vezérlés leírása



Szoftver Engineering

Szoftver fejlesztés

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



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 és a minőség

- Tervezés minősége alapvetően befolyásolja a végtermék minőségét
 - felhasználói követelmények → rendszer
 - összes követelmény lefedése
 - érthetőség
 - teljesség: adat, funkció, viselkedés
 - minőségi kritériumrendszerek



Strukturált tervezés menete

- tipikusan iteratív folyamat
 - absztrakt reprezentáció → konkrét reprezentáció
- tervezési modell
 - teljes reprezentációja a szoftvernek
 - különböző nézetek (aspektusok)



Általános tervezési elvek



Általános tervezési elvek

- 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

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



Általános tervezési elvek

- 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
- ellenőrizni kell a tervet a szemantikus hibák kijavítása érdekében
 - ellentmondások, redundancia...
 - nem csak szintaktikusan kell ellenőrizni

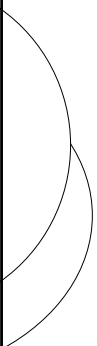


Tervezési módszerek



Tervezési módszerekről általában

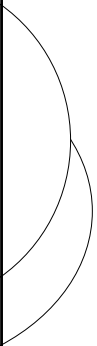
- Módszerek:
 - segítség a döntéseknél, de...
- Kérdések tervezéskor:
 - Mi alapján lehet a szoftvert részekre osztani?
 - Hogyan legyenek a funkciók, ill. a adatszerkezetek részletei elválasztva a szoftver koncepcionális modelljétől?
 - Van-e általánosan használható mérőszám a tervezés minőségének mérésére?



Tervezési módszerek - Absztrakció I.

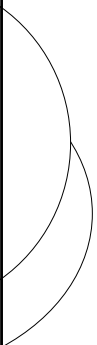
- Absztrakció
 - absztrakt: megoldás (modell) a probléma tér fogalmaival
 - kevésbé absztrakt: keverednek a probléma tér fogalmi az implementációs tér fogalmaival
 - közvetlenül megvalósítható megoldás leírás

- tervezés során az absztrakció csökken



Tervezési módszerek – Absztrakció II.

- Absztrakció típusai
 - működés leírás (procedural) absztrakció
 - függvény név – függvény utasítások
 - adat absztrakció
 - adat szerkezet név – adat szerkezet definíció
 - vezérlés absztrakció
 - szemafor



Tervezési módszerek – Pontosítás, részletezés

- (refinement)
- top-down tervezési modell
- elsősorban működés leírás kidolgozására
- lépésenkénti finomítása, részletezése a leírásnak

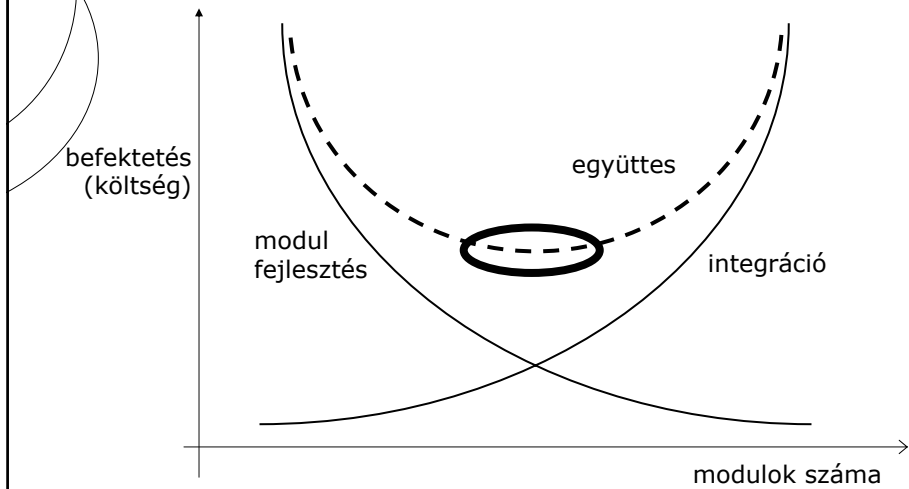
Tervezési módszerek – Modulokba szerverés

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

Modulok méretének meghatározása

- Probléma: p_1, p_2
- Komplexitás: $C(p_1), C(p_2)$
- Befektetés: $B(p_1), B(p_2)$
- $C(p_1) > C(p_2) \rightarrow B(p_1) > B(p_2)$
- $C(p_1 + p_2) > C(p_1) + C(p_2) \rightarrow$
 $B(p_1 + p_2) > B(p_1) + B(p_2)$
- \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 felépítés, struktúrája
- komponensek (nem meghatározott a méretük) hierarchiája
- komponensek együttműködésének módja

- koncepcionális leírás



Szoftver architektúra leírása

- strukturális modell
- framework modell
 - absztraktabb, általános működés hasonló rendszerekben, tervezési minták
- dinamikus modellek
 - viselkedés (a rendszer hogyan változik külső hatásokra)
- folyamat modellek
 - üzleti, technológia
- funkcionális modellek
 - hierarchia
- leírása: Architectural Description Language



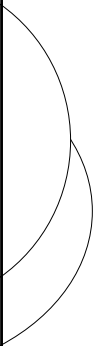
Strukturális modell

- Struktúra
 - Vezérlési hierarchia
 - Adat elemek 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
 - vezérlő és munkavégző modulok
 - változás általában a munkavégző modulokban → karbantarthatóbb (kevesebb mellékhatás)



Adat elemek hierarchiája (struktúra)

- alap adat típusok
- adatszerkezetek:
 - vektorok
 - több dimenziós tömbök
 - láncolt listák
- hierarchikus adatszerkezetek



Általános elvek 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
- Adat folyam (data flow)
 - pipe, batch ...
- „Call and return” architektúra
 - program – alprogram
 - remote procedure call (kliens-szerver)
- Objektum orientált architektúra
- Rétegszerkezet



Strukturált architektúra tervezés




Adat modellezés

- Entitás-relációs diagram
 - adat objektum – entitás
 - belső reprezentáció
 - attribútumok
 - entitás rendszer által kezelt tulajdonságai
 - kapcsolatok
 - modalitás
 - kötelező, opcionális
 - számosság



Entitás-relációs diagram kiterjesztése

- Entitások hierarchikus viszonya
 - ~ fa struktúra
 - entitások minősítése (~ attribútumok, kategorizálás, dimenzió)
 - kapcsolati modell
 - részek definiálása



Funkció és információ áramlás modellezése

- Információ folyam leírás
- Adat folyam (data flow) diagram
- Lépés viselkedés leírás felé
 - vezérlés intenzív működés
 - idő kritikus műveletek
 - kiterjesztett adat folyam diagram
- Viselkedés leírás
 - állapot átmenet diagram



Strukturált tervezés fázisai (volt)

- **Architektúra kialakítása:**
 - Adat tervezés
 - adat könyvtár, ER → adat struktúrák
 - Architektúra tervezés
 - strukturális felépítése a rendszernek
 - → minták, specifikáció, részek...
- **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)



Objektum orientált tervezés



Objektum orientált tervezés

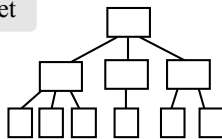
- Vezérlési hierarchia (struktúra) és adat hierarchia (struktúra) együtt
- Működő rendszer: együttműködő objektumok halmaza
- Objektum: adatok és műveletek (metódusok) egysége
 - adat ~ tulajdonság
 - műveletek (metódusok) üzenettel aktivizálható működés
 - objektum felelősségei

Objektum orientált tervezés

- Rendszer tervezése, modellezés, gondolkodás objektum orientált módon
- Rendszer fejlesztés támogatása:
 - objektum orientált modellező, tervező, fejlesztő rendszerek (CASE)

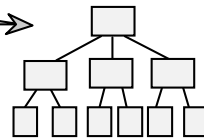
Vezérlési hierarchia és adat hierarchia együttes meghatározása

adatszerkezet

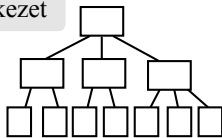


Az OO megközelítés a feladatot egyetlen egységes módon bontja fel.

objektumstruktúra

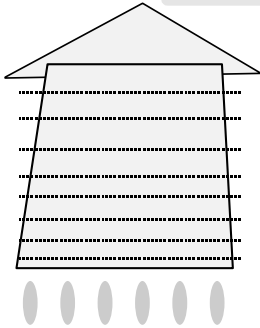


vezérlési szerkezet



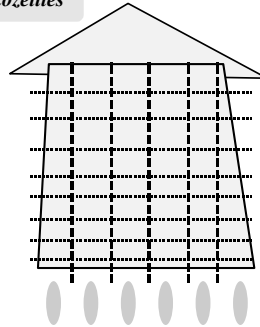
Objektum orientált megközelítés egyik előnye

Strukturált szemlélet



A funkcionális dekompozíció a folyamatnak csak az „időbeli” felosztását képes kifejezni.

OO megközelítés

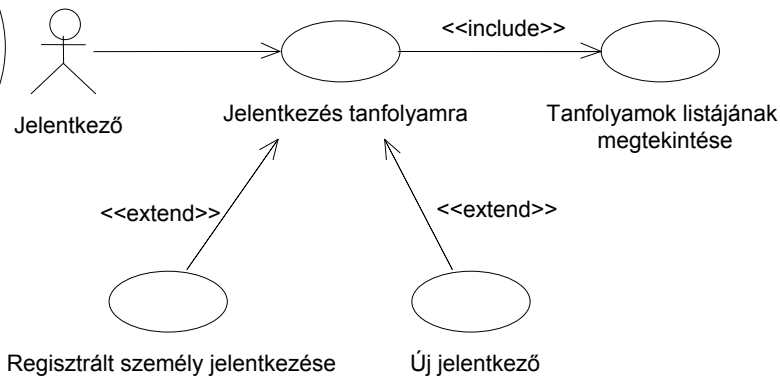


A módszerekkel az összetett folyamat „térbeli”, azaz objektumokhoz rendelt felosztása is kifejezhető.

Objektum orientált modellezés

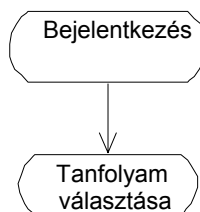
- Use case (használati eset) diagram
 - nem kizárólag OO
- Aktivitás diagram
- Szekvencia diagram
- Együttműködési (Collaboration) diagram
- Osztály diagram
- Állapot-átmenet diagram
- ...

Use case (használati eset) diagram



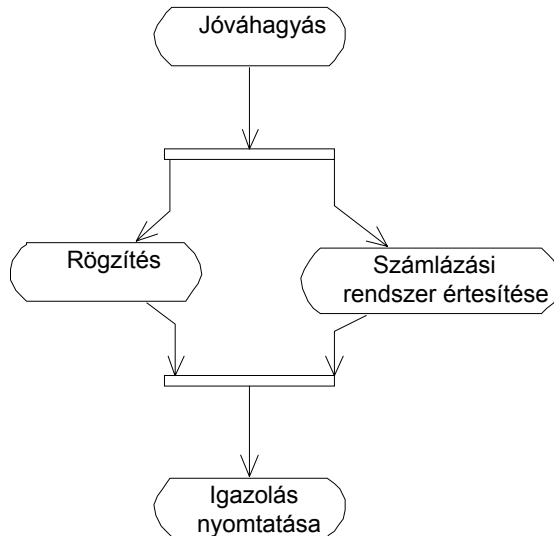
Aktivitás diagram: tevékenységek

- Aktivitás, tevékenység (activity)
 - Valamilyen tevékenység, amit meg kell csinálni



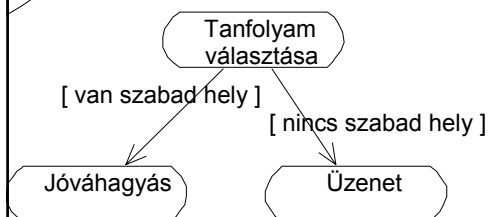
- Szekvencia: a tevékenységet egy másik tevékenység követ

Aktivitás diagram: párhuzamos tevékenységek

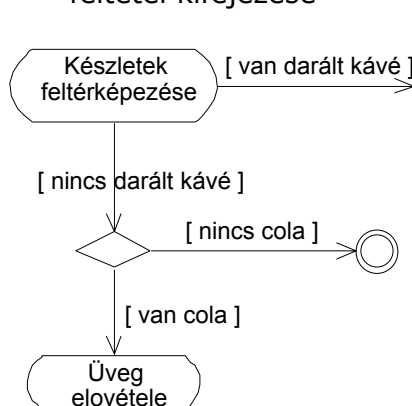


Aktivitás diagram: Döntés

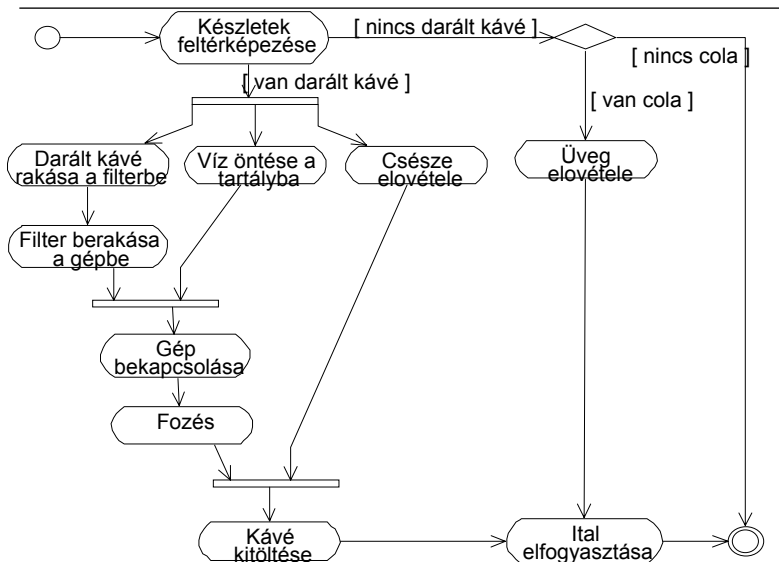
- o Egyetlen feltétel definiálása az átmenethez



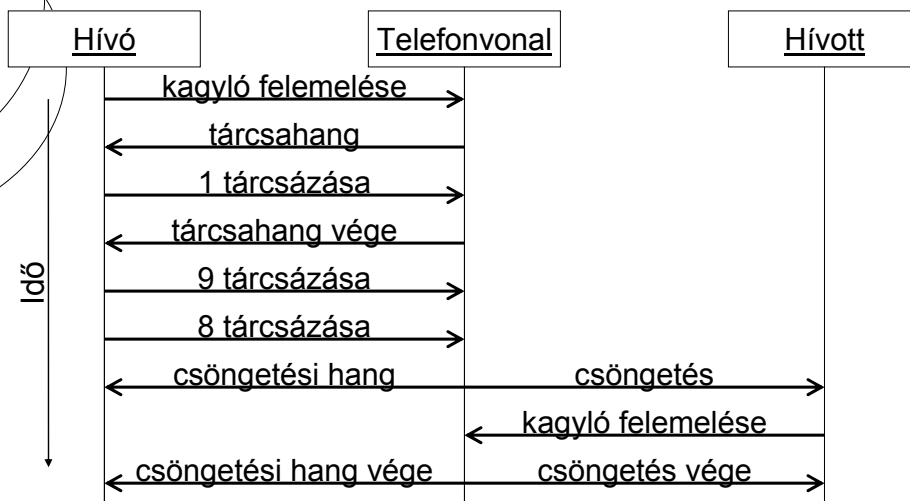
- o Döntés: több egymásba ágyazott feltétel kifejezése



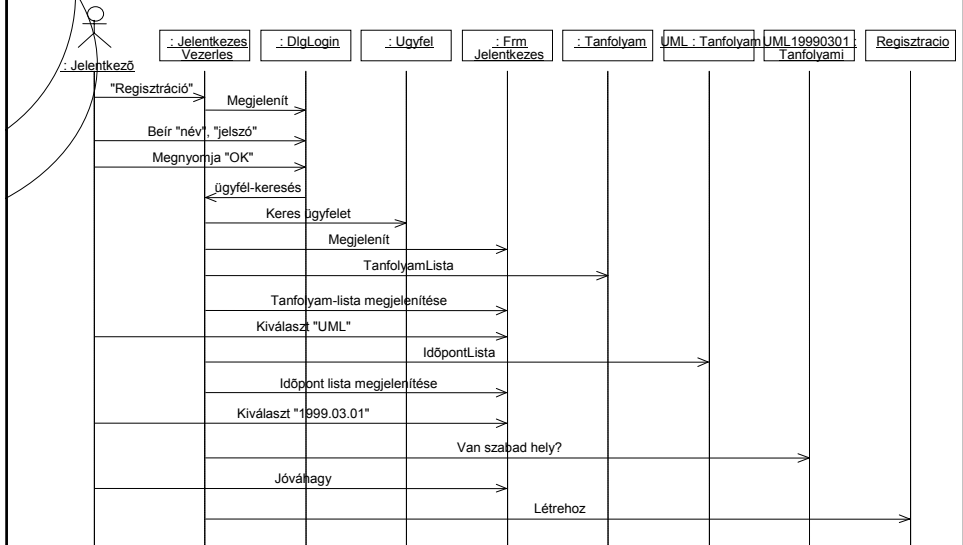
Aktivitás diagram példa



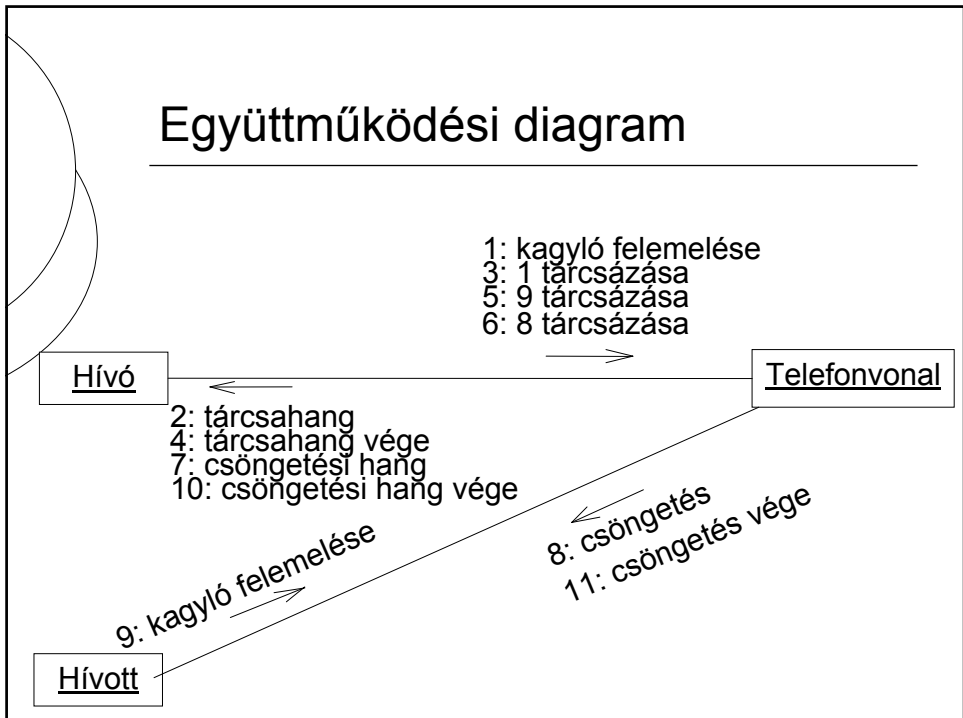
Szekvencia diagram



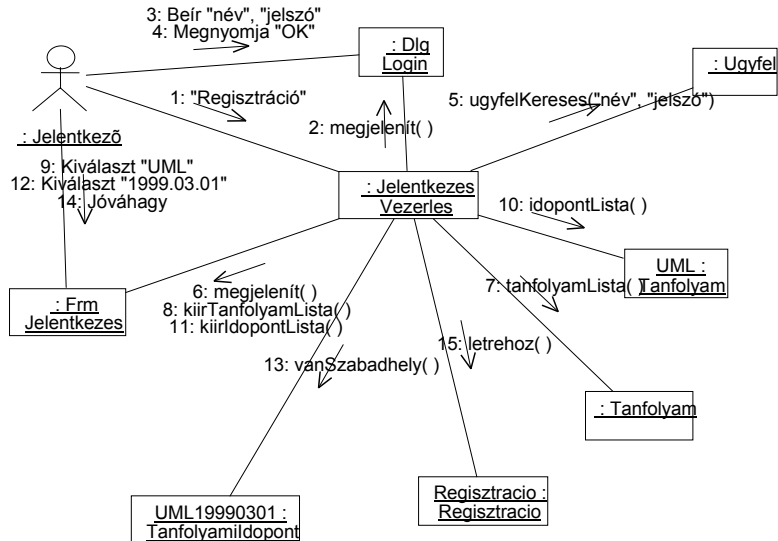
Szekvencia diagram



Együttműködési diagram



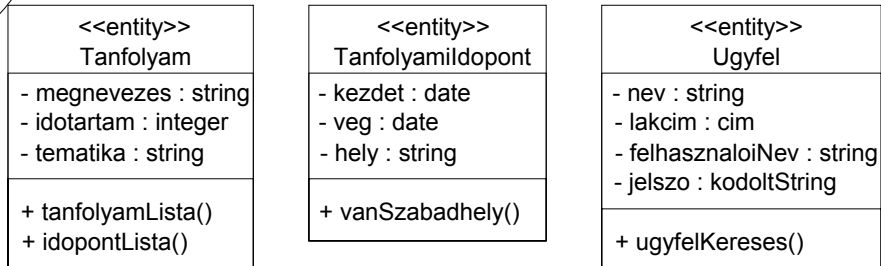
Együttműködési diagram



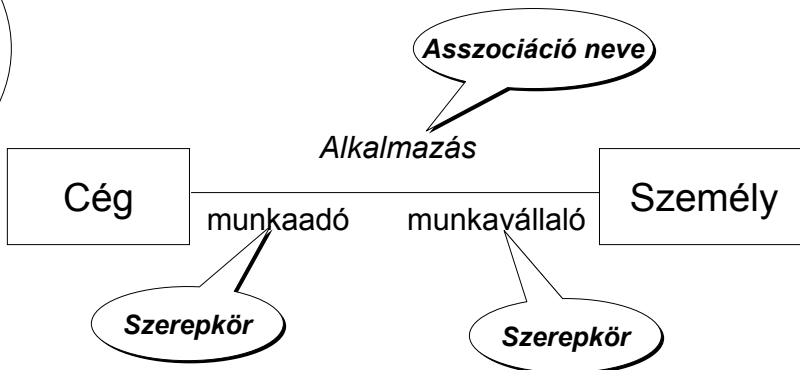
Osztály diagram

- osztályok
- kapcsolatok
- öröklés
- ...

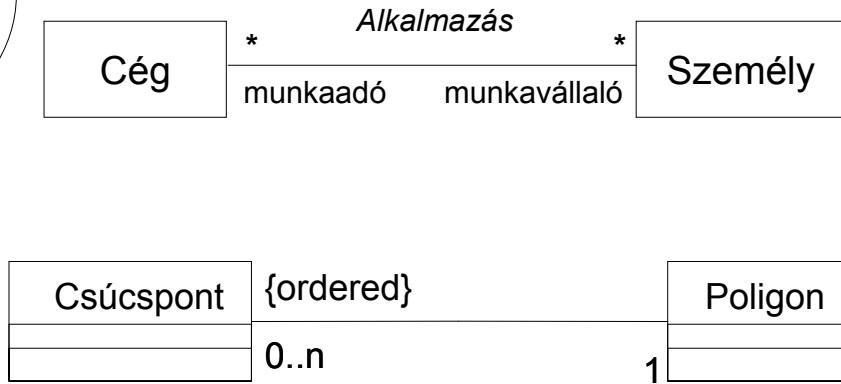
Osztály diagram: osztályok



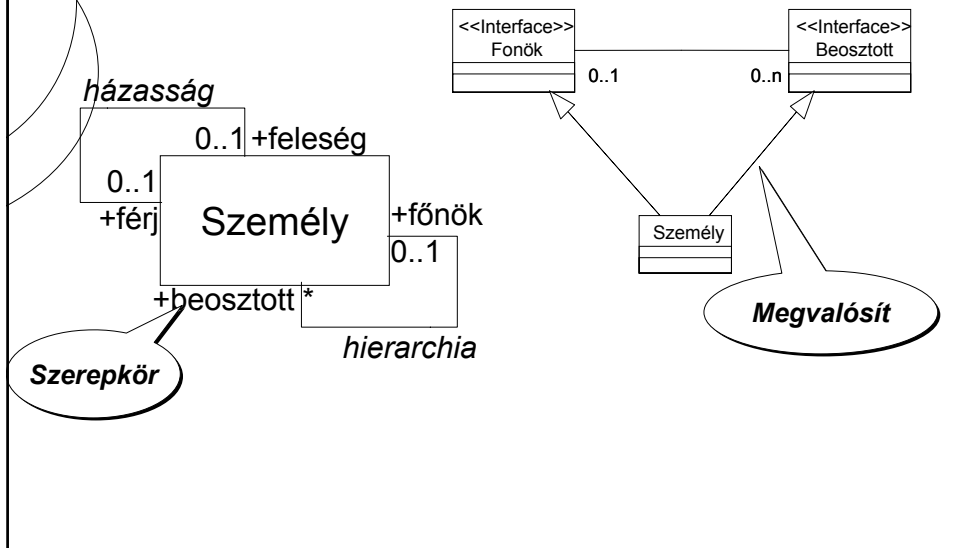
Osztály diagram: kapcsolatok



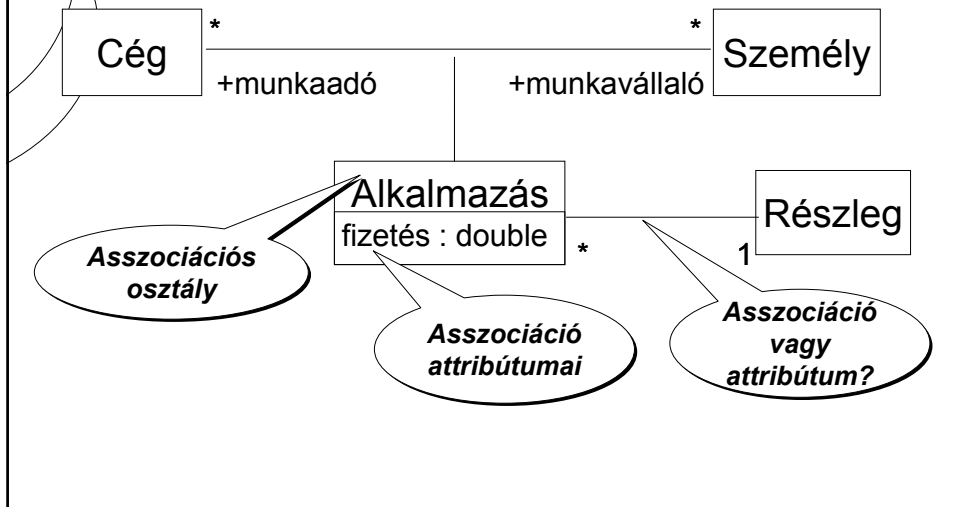
Példa kapcsolatokra



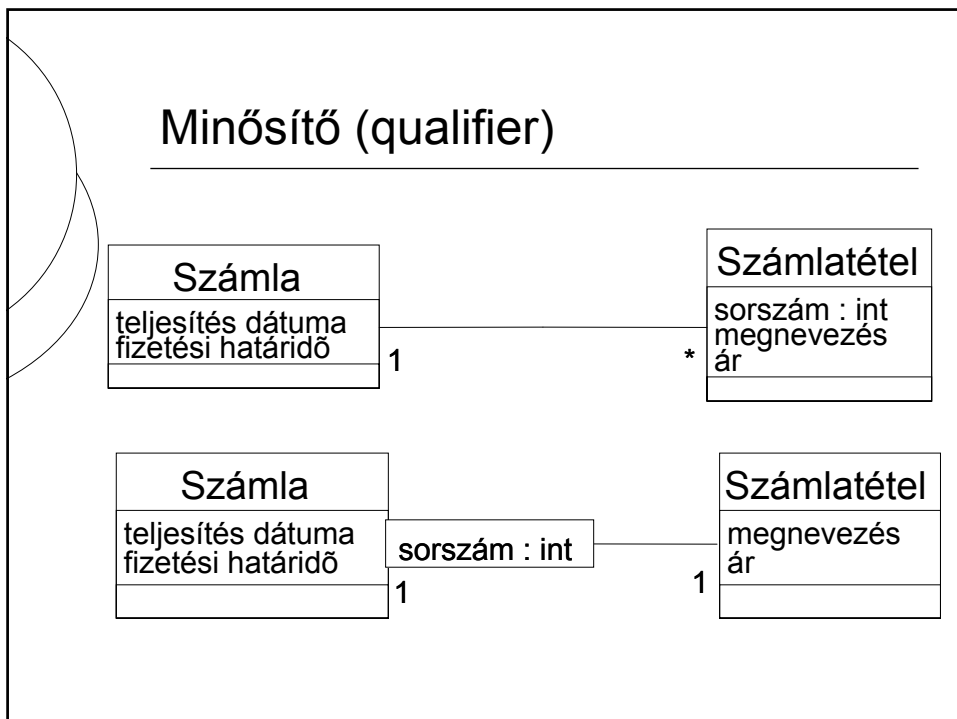
Osztálynak önmagával való asszociációja (self-association)



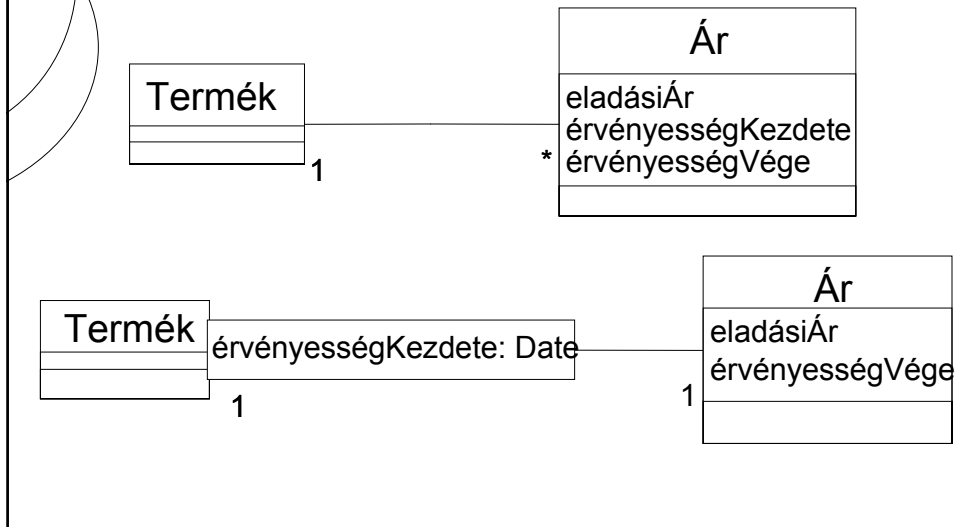
Asszociációs osztály



Minősítő (qualifier)



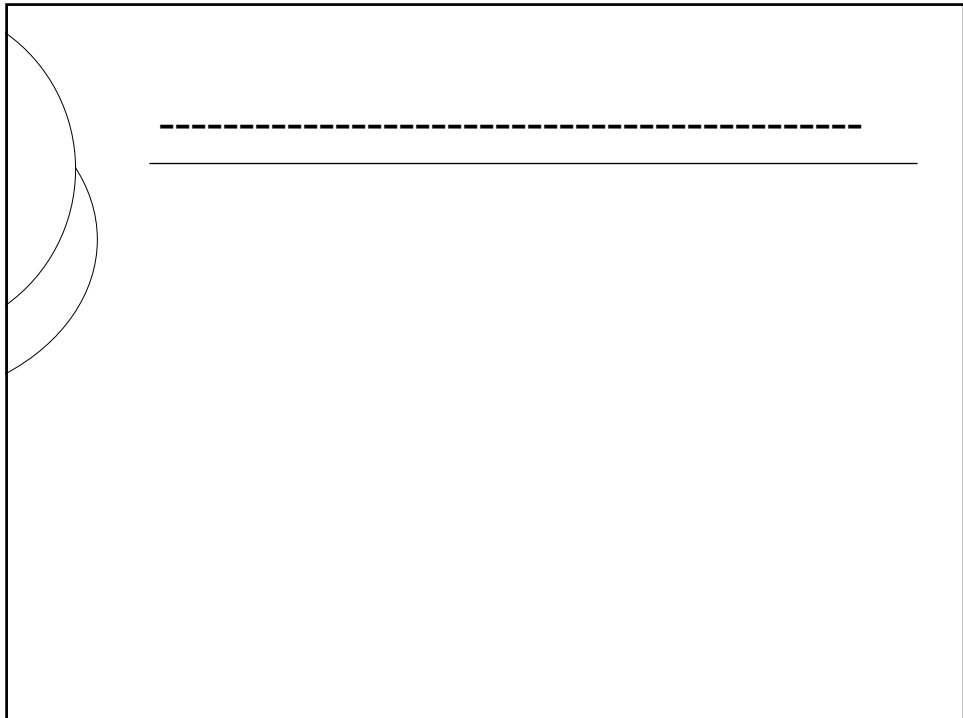
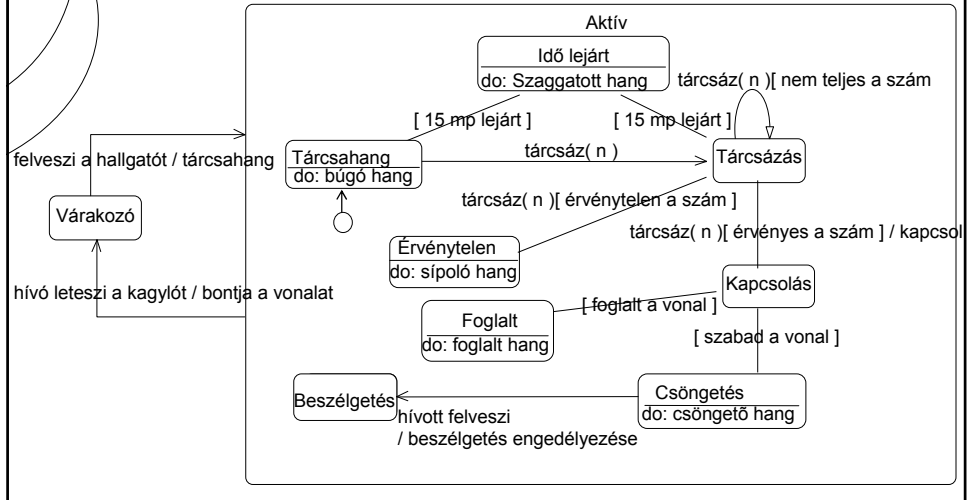
Minősítő (qualifier)



Szekvencia diagram (sequence diagram)

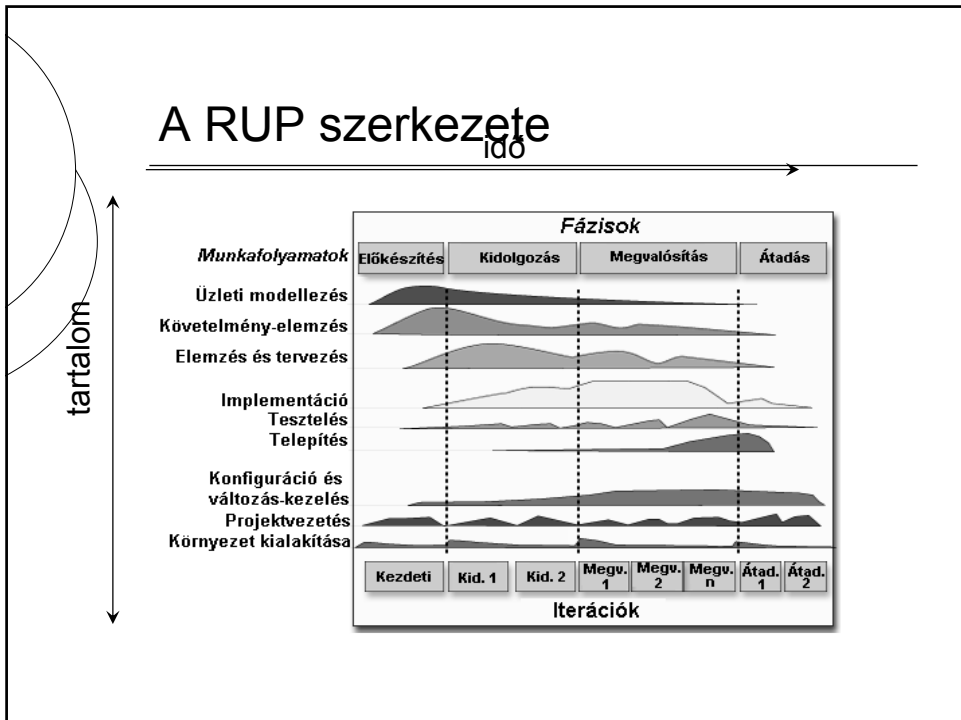
- Az adott folyamat egy konkrét végrehajtását írja le az objektumok közötti kommunikáción keresztül

Állapot-átmenet diagram



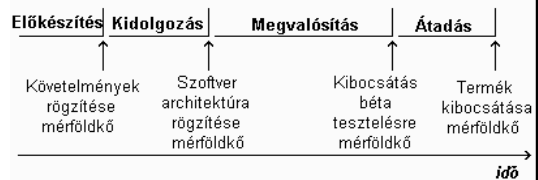
REZERVÁTUM

A RUP szerkezete

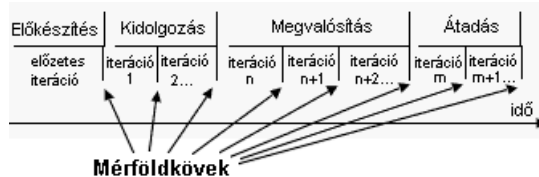


Rational Unified Process

- A Rational Unified Process a szoftverfejlesztés életciklusát négy egymást követő fázisra bontja:
 - Előkészítés (Inception)
 - Kidolgozás (Elaboration)
 - Megvalósítás (Construction)
 - Átadás (Transition)
- Minden fázis végén jól-definiált mérföldkövek vannak: kritikus döntéseket kell hozni
 - Értékelni az eddigi eredményeket
 - Dönteni a folytatásról



Dinamikus aspektus - fázisok és iterációk



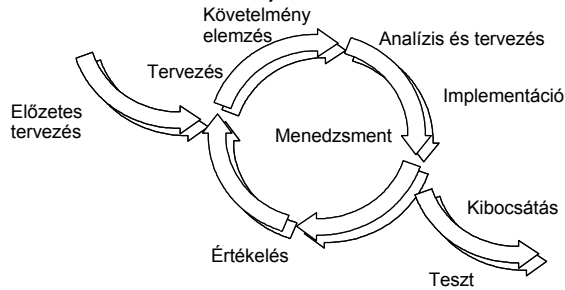
Az aktuális feladat dönti el, hogy hány iterációra van szükség a feladat elvégzéséhez.

Az iterációk tervezése kritikus feladat a projekt tervezése során.

RUP - szemlélet

o Iteratív fejlesztés

- Pontos projekt vezetői követést igényel
- Hibát követünk el ha nem vesszük komolyan
- Felhasználó folyamatos bevonása
- Elefántcsont torony kiiktatása



RUP - kockázat csökkentése

