





# ***A rendszerfejlesztés technológiája és módszertana***

*Dancs Sándor*  
*Nyíregyházi Egyetem*  
*Matematika és Informatika Intézet*





# 1. Bemutatózás, a tananyag beosztásának áttekintése és a követelmények ismertetése

## Kurzusinformáció





Köszönöm a figyelmet!





## 2. A rendszerfejlesztés élelciklusa



# Szoftver





# Szoftverfolyamat

- Szoftverspecifikáció
- Szoftvertervezés és implementáció
- Szoftvervalidáció
- Szoftverevolúció



# Szoftverspecifikáció (követelménytervezés)


- Megvalósíthatósági elemzés
- Követelmények feltárása és elemzése
- Követelmény specifikáció
  - Felhasználói követelmények
  - Rendszerkövetelmények
- Követelmény validáció



# Szoftvertervezés és implementáció

- Architektúrális tervezés
- Objektum-orientált tervezés





# Szoftvervalidáció (verifikáció és validáció)

- Komponens tesztelése
- Rendszer tesztelése
- Átvételi tesztelés



# Szoftverevolúció (szoftverkarbantartás)

Szoftver továbbfejlesztése a változó elvárásoknak megfelelően



Köszönöm a figyelmet!





# Szoftverspecifikáció (követelménytervezés)



# 3. Követelmények meghatározása

# Követelménytervezés fázisai

- Megvalósíthatósági elemzés
- Követelmények feltárása és elemzése
- Követelmény specifikáció
  - Felhasználói követelmények
  - Rendszerkövetelmények
- Követelmény validáció



# Megvalósíthatósági elemzés

A rendszer

- kivitelezhető-e és
- költséghatékony-e?

Eredmény: megvalósíthatósági jelentés

# Követelmények feltárása és elemzése

- A terület megfigyelése, megismerése
- Követelmények összegyűjtése (együttműködés a megrendelőkkel, felhasználókkal)
- Csoportosítás
- Ellentmondások kezelése
- Követelmények rendezése (fontossági sorrend)
- Követelmények ellenőrzése





# Követelmény specifikáció

## Követelmények típusai

- Funkcionális követelmények
- Nemfunkcionális követelmények
  
- Követelménytípusok még
  - Felhasználói követelmények
  - Rendszerkövetelmények



# Funkcionális követelmények

„Mit csináljon a rendszer”

A szoftverrel szemben támasztott elvárás



# Nemfunkcionális követelmények

„Hogyan csinálja a rendszer”

- Teljesítmény
- Megbízhatóság
- Használhatóság
- Hordozhatóság



# Felhasználói követelmények

„A rendszer viselkedését írjuk le, azt, hogy mit csinál, nem azt, hogy hogyan”

Leírása: egyszerűen, táblázatok, diagrammok segítségével

# Rendszerkövetelmények

„Azt írjuk le, hogy a rendszernek hogyan kell biztosítania a felhasználói követelményeket”

Leírása: részletesen, egyértelműen, pontos jelölésekkel, a tervezési és megvalósítási megoldások mellőzésével



# Követelmény validáció

„Azt ellenőrizzük, hogy a követelmények valóban azt a rendszert írják le, amit a megrendelő szeretne”

Cél: a hibák megtalálása és kijavítása





# Követelmények dokumentálása

Végeredmény: a felhasználói és a rendszer-  
követelmények részletes specifikációja



Köszönöm a figyelmet!







# 4. Folyamatmodellek, rendszermodellek



# A szoftverfolyamat modelljei

- Vízésésmodell
- Evolúciós fejlesztés
- Újrafelhasználásra épülő (komponensalapú) fejlesztés
- Iteratív (inkrementális és spirális) fejlesztés



# Vízésésmodell

- Szoftverspecifikáció
- Szoftvertervezés
- Implementáció
- Szoftvervalidáció
- Szoftverevolúció



# Evolúciós fejlesztés

- Feltáró fejlesztés
- Prototípus készítés



# Újrafelhasználásra épülő (komponensalapú) fejlesztés

- Komponensek elemzése
- Követelmények módosítása
- Tervezés komponensek újrafelhasználásával
- Komponensek fejlesztése és rendszerbe integrálása



# Iteratív fejlesztés

- Inkrementális fejlesztés
- Spirális fejlesztés



Köszönöm a figyelmet!





# 5. UML

(Unified Modeling Language,  
Egységes Modellezési Nyelv)

[uml.org](http://uml.org)



# Adattagok, tagfüggvények

kötőjel: private  
pluszjel: public  
kettőskereszt: protected  
hullámvonal: package private  
aláhúzás: static

<b>Osztály neve (dőlt ha, absztrakt)</b>
<u>- adattag: Típus</u>
# adattag: Típus
<u>+ függvény(paraméter: Típus, ...): Típus</u>
~ függvény(paraméter: Típus, ...): Típus



Asszociáció



Aggregáció



Kompozíció



Öröklődés



Függőség



Köszönöm a figyelmet!






# Szoftvertervezés és implementáció



„A szoftver logikai szerkezetére vonatkozó döntéseket hozzuk meg.

A logikai szerkezetet egy modellezőnyelv (UML) segítségével ábrázoljuk”





# 6. Architektúrális tervezés, objektum-orientált tervezés



# Architekturális tervezés

„A rendszereket alrendszerekre bontjuk.

Cél a funkcionális és nem funkcionális rendszerkövetelményeknek megfelelő rendszerszerkezet létrehozása”








# Modellek

- Tárolási
- Kliens-szerver
- Rétegzett



# Az alrendszerrek modulokra való bontásának stratégiái

- Objektumorientált
- Funkcióorientált




# Vezérlési stílusok

- Központosított
- Eseményalapú



# Objektum-orientált tervezés

„Az objektumosztályokat és a köztük lévő kapcsolatokat tervezzük meg”





# Objektumorientált stratégia

- Objektumorientált elemzés
- Objektumorientált tervezés
- Objektumorientált programozás

# Objektumorientált paradigma

- Objektum
  - Állapot (tulajdonság(ok), azaz változó(k))
  - Viselkedés (metódus(ok), azaz függvény(ek))
- Osztály
  - Hasonló objektumok közös változóit és metódusait leíró sablon

Az objektum egy osztály példánya!

Az osztály az objektum típusa!

# Egységbe zárás (encapsulation)

A változók és a metódusok osztályba való becsomagolása

# Információrejtés

Az objektum belső működésének elrejtése a külvilág elől

Ennek megfelelően az osztály két részből áll:

- Publikus interfész (mire jó, hogyan használható)
- Implementáció (hogyan működik)





Köszönöm a figyelmet!






# 7. 1. zárthelyi dolgozat



Köszönöm a figyelmet!





# 8. Felhasználói felületek tervezése, ergonómia Felhasználói támogatás, dokumentáció

## Gyakorlat






Köszönöm a figyelmet!





# 9. Verifikáció és validáció



# Szoftvervalidáció (verifikáció és validáció)

Verifikáció: „Azt ellenőrizzük, hogy a szoftver megfelel-e a specifikációnak”

Validáció: „Azt ellenőrizzük, hogy a szoftver megfelel-e, a megrendelő elvárásainak”



# Szoftvertesztelés

- Komponens tesztelése
- Rendszer tesztelése
- Átvételi tesztelés





# Szoftverevolúció (szoftverkarbantartás)

Szoftver továbbfejlesztése a változó elvárásoknak megfelelően



Köszönöm a figyelmet!





# 10. Tesztelőeszközök

## Projektmenedzselés, PRINCE metodika

### Gyakorlat





Köszönöm a figyelmet!





# 11. Minőségbiztosítási szabványok

## Verziókezelő rendszerek, SVN

### Gyakorlat





Köszönöm a figyelmet!





# 12. RUP, SSADM



# RUP folyamatmodell (Rational Unified Process, Rational egységes fejlesztési módszer)






# Rendszerfejlesztés folyamata

- Dinamikus (modell fázisai)
- Statikus (végrehajtandó folyamattevékenységek)
- Gyakorlati (javasolt gyakorlatok)




# Rendszerfejlesztés fázisai

- Előkészítés
- Kidolgozás
- Megvalósítás
- Átadás



# SSADM

(Structured Systems Analysis and  
Design Method, Strukturált  
Rendszerelemzési és Tervezési  
Módszertan)





# SSADM

## Műszaki és dokumentációs szabványok gyűjteménye

### Felülről-lefelé haladó elemzési és tervezési módszer





# SSADM két fő részből áll

- Felhasználói követelmények elemzése
- Rendszer tervezése



Köszönöm a figyelmet!





# 13. 2. zárthelyi dolgozat



Köszönöm a figyelmet!







# 14. Fejlesztői céghez látogatás, tapasztalatgyűjtés



Köszönöm a figyelmet!

