

Dr. Mileff Péter

Szoftverfejlesztés

Bevezetés

Miskolci Egyetem
Általános Informatikai Tanszék

Bemutkozás és követelmények

- Dr. Mileff Péter - Általános Informatikai Tanszék
 - Fizika Tanszék – A/1 - 303. szoba.
 - Konzultációs idő: ???.
- Követelmények:
 - Vezetett gyakorlat nincs.
 - Jelenléti ív nincs.
 - Zárthelyi dolgozat nincs.
 - Féléves feladat van: Konzultációs jellegű
 - Előadás tartása egy előre megbeszélt témából.
 - Zárás:
 - Aláírás + kollokvium (írásbeli és szóbeli)

Jegyzet: <http://www.iit.uni-miskolc.hu/~mileff/>

2

A szoftver

- A *szoftver* szót sokan egyenlőnek tekintik a számítógépes programokkal.
 - Nincs egyértelmű definíciója.
- Több ennél:
 - hozzájuk kapcsolódó dokumentációk,
 - konfigurációs adatok.
- Ezek elengedhetetlenek ahhoz, hogy ezek a programok helyesen működjenek.

3

Szoftvertermékek csoportjai

- Általános termék:
 - egyedülálló rendszerek, egy fejlesztő szervezet készíti, és adja el. **Dobozos szoftverek.** Pl.: adatbázis-kezelők, szövegszerkesztők.
- Egyedi igényekre szabott (rendelésre készített) termékek
 - Egyéni megrendelők megbízásai alapján készülnek speciális megrendelői igények alapján.
 - Pl.: az elektromos eszközök vezérlőrendszerei, a forgalomirányító és ellenőrző rendszerek
- A határ gyakran összemosódik.

4

A szoftverfolyamat...

5

A szoftverfolyamat

- **Tevékenységek és eredmények sora, amelyek egy szoftvertermék előállításához vezetnek.**
- Komplex, és nagyban függ az emberi tevékenységektől:
 - Ezért nem igazán automatizálható CASE (számítógéppel segített szoftvertervezés) eszközökkel.
 - Nincs ideális, minden számára megfelelő folyamat.
 - A szoftver fejlesztés minden szervezetnél más!
- **Cél:** úgy kell kialakítani, hogy kiaknázzák a szervezeten belül az emberek képességeit és a fejlesztő rendszer jellegzetességeit.

6

A folyamat közös fázisai

- **Szoftverspecifikáció:** A szoftver funkcióit, illetve annak megszorításait definiálni kell.
- **Szoftvertervezés és implementáció:** A specifikációnak megfelelő szoftvert elő kell állítani.
- **Szoftvalidáció:** a szoftvert validálni kell, hogy biztosítsuk, azt fejlesztettük, amit az ügyfél kíván.
- **Szoftverevolúció:** A szoftvert úgy kell kialakítani, hogy megfeleljen a megrendelő kívánsága szerint történő változtatásoknak.

7

A szoftverfolyamat modelljei

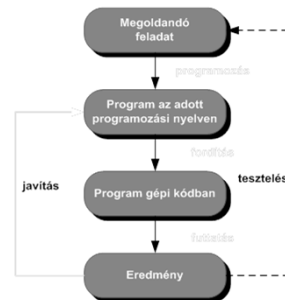
- **A szoftverfolyamat absztrakt reprezentációja:**
 - speciális perspektívából reprezentál egy folyamatot
 - Részleges információk a folyamatról, mert általánosak.
- Ismertebb modellek:
 - **Vízesésmo­dell:** Ez a folyamat alapvető tevékenységeit a folyamat különálló fázisainak tekinti.
 - **Evolúciós vagy iteratív fejlesztés:** összefésüli a specifikáció, a fejlesztés és a validáció tevékenységeit.
 - **Komponens alapú fejlesztés:** nagy mennyiségű újrafelhasználható komponensek létezésén alapszik.
- A gyakorlatban keveredhetnek egymással.

8

Ismertebb modellek...

9

Egyszerű programfejlesztési modell



10

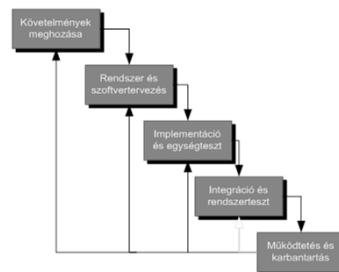
Egyszerű programfejlesztési modell

- A kis programok létrehozásának a modellje
- Általában egyszemélyes programfejlesztésnél használjuk
- **Oka:**
 - a megoldandó feladat könnyen áttekinthető és modellezhető,
 - a probléma azonnal megfogalmazható egy adott programozási nyelven
- A futási eredményeket a feladattal vetjük egybe
- A javításokat közvetlenül a programozási nyelvű leírásban, a programkódban hajtjuk végre.

11

A vízésésmodell

- A szoftverfejlesztés folyamatának első publikált modellje, más tervezői modellekből származik.



12

1 fázis: követelmények elemzése és meghozása

- A rendszer felhasználóival való konzultáció alapján kialakul a:
 - rendszer szolgáltatásai,
 - megszorításai,
 - célja.
- } specifikáció
- Ezek részletes kifejtése szolgáltatja rendszer specifikációját.

13

2 fázis: rendszer - és szoftvertervezés

- A tervezési folyamatban szétválasztódik a hardver és szoftverkövetelmény.
- A rendszer átfogó architektúráját itt kell kialakítani.
 - Milyen modell?
 - Milyen alrendszerek, azok kapcsolata.
- Szoftverrendszer-absztrakciók és a köztük lévő kapcsolatok tervezése és leírása.

14

3 fázis: implementáció és egységteszt

- Ebben a szakaszban megvalósul a szoftverterv (annak részei)
 - programok, illetve programegységek (komponensek) halmazaként.
- Az egységteszt azt ellenőrzi, hogy minden egység megfelel-e a specifikációjának.

15

4 fázis: integráció és rendszerteszt

- A különálló programegységek, programok integrálása.
- Teljes rendszerként való tesztelése.
- **Cél:** annak megállapítása, hogy **a rendszer megfelel-e a követelményeknek.**
- A tesztelés után a szoftverrendszer átadható az ügyfélnek.

16

5 fázis: Működtetés és karbantartás

- Általában a szoftver életciklusának leghosszabb fázisa.
- Beletartozik:
 - A később kiderült hibák javítása.
 - A rendszeregységek implementációjának továbbfejlesztése.
 - Új követelmények léphetnek fel, így szükséges lehet a rendszer szolgáltatásainak továbbfejlesztése.

17

Áttekintés

- A fázisok eredménye egy dokumentum.
- Egy fázis csak akkor indulhat, ha az előző befejeződött.
- A folyamat **nem egyszerű lineáris modell**, hanem a fejlesztési tevékenységek iterációjának sorozata.
 - Az iterációk költségesek, így gyakran befagyasztják őket. A problémák megoldása később, vagy soha. **Hátránya:** a szoftver nem azt csinálja, amit elvárnak tőle.
- Csak akkor használható jól, ha már előre jól ismerjük a követelményeket.

18

EVOLÚCIÓS FEJLESZTÉS...

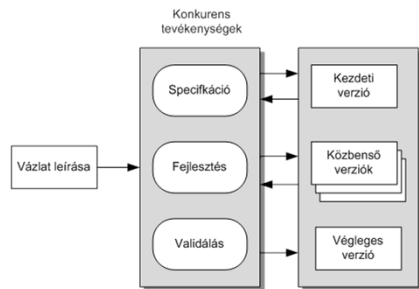
19

Evolúciós fejlesztés

- **Alapötlete:**
 - a fejlesztőcsapat kifejleszt egy kezdeti implementációt,
 - majd azt a felhasználókkal véleményeztetni,
 - végül sok-sok verzió keresztül addig finomítani, amíg a megfelelő rendszert el nem érjük.
- A megközelítési mód sokkal jobban érvényesíti a tevékenységek közötti párhuzamosságot és a gyors visszacsatolásokat.

20

Evolúciós modell



21

A két típusa

- **1. Feltáró fejlesztés:**
 - **Cél:** a megrendelővel együtt tárjuk fel a követelményeket, és alakítsuk ki a végleges rendszert.
 - A fejlesztés a rendszer már ismert részeivel kezdődik. A végleges rendszer úgy alakul ki, hogy egyre több, az ügyfél által kért tulajdonságot társítunk a már meglévőkhöz.
- **2. Eldobható prototípus készítés:**
 - **Cél:** a lehető legjobban megértsük az ügyfél követelményeit, amelyekre alapozva pontosan definiáljuk azokat.
 - A prototípusnak pedig azon részekre kell koncentrálni, amelyek kevésbé érthetők.

22

Áttekintés

- **Előnye:**
 - hatékonyabb a vizesésmodellnél, ha olyan rendszert kell fejleszteni, amely közvetlenül megfelel az ügyfél kívánságainak.
 - a rendszerspecifikáció inkrementálisan fejleszthető.
- **Hátránya a vezetőség szemszögéből:**
 - **A folyamat nem látható:** a menedzsereknek szüksége van a részeredményekre. (Fejlődés mérése)
 - **A rendszerek gyakran szegényesen strukturáltak:** A folyamatos változtatások lerontják a rendszer struktúráját.
- Rövid élettartalmú, kis és közepes rendszerek esetén célszerű alkalmazni. (~500000 programsor)

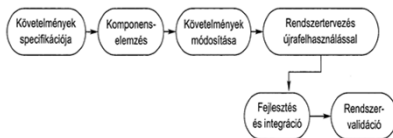
23

KOMPONENS ALAPÚ FEJLESZTÉS...

24

Komponens alapú fejlesztés

- Alapgondolata az újrafelhasználható komponensekből való építkezés.
 - A szoftverfolyamatokban megtalálhatók a komponensek újrafelhasználása:
 - Korábbi kód átdolgozása, felhasználása, általánosítása.



25

Komponenselemzés

- A követelményspecifikáció alapján komponensek keresése, hogy melyek implementálják azokat.
 - Mely kódok használhatók újra fel?
- Általában nincs egzakt illeszkedés, a felhasznált komponens a funkciók csak egy részét nyújtja.

26

Követelménymódosítás

- A követelmények elemzése a megtalált komponensek információi alapján.
 - Módosítás az elérhető komponenseknek megfelelően.
- Ahol a módosítás nem lehetséges, ott újra el kell végezni a komponenselemzést
 - alternatív megoldást kell keresni,
 - vagy új komponens kifejlesztésének indítványozása.

27

Rendszertervezés újrafelhasználással

- Ez a szakasz felelős a rendszer szerkezetének **tervezéséért**:
 - Számba kell venni:
 - hogy milyen komponenseket akarnak újrafelhasználni,
 - melyeket kifejleszteni, vagy beszerezni,
 - egy logikus, áttekinthető szerkezetet kialakítani, hogy azok működhessenek.
 - Ha nincs elérhető újrafelhasználható komponens:
 - új szoftverek is kifejleszthetők,
 - vagy megvásárolhatók.

28

Fejlesztés és integráció

- 1. A nem megvásárolt komponenseket ki kell fejleszteni és a rendszerbe integrálni.
 - Tervezés szükséges.
- 2. Az átalakítandó komponenseken a szükséges módosításokat elvégezni.
 - Módosítás, általánosítás, stb.
- A rendszer-integráció ebben a modellben sokkal inkább a fejlesztési folyamat része, mint különálló tevékenység.

29

Áttekintés

- **Előny:**
 - csökkenti a kifejlesztendő szoftverek számát
 - Ezzel csökkenti a költségeket, és
 - a kockázati tényezőket.
 - A rendszer így gyorsabban leszállítható sok esetben.
- **Hátrány:**
 - a követelményeknél elkerülhetetlenek a kompromisszumok.
 - Következménye: a rendszer nem felel meg a felhasználó valódi kívánságainak.

30

FOLYAMAT-ITERÁCIÓ

1. INKREMENTÁLIS FEJLESZTÉS
2. SPIRÁLIS FEJLESZTÉS

31

Folyamat - iteráció

- A szoftverfolyamat nem egy egyszerű folyamat:
 - a folyamattevékenységek rendszeresen ismétlődő folyamata.
 - a rendszert mindig átdolgozzuk az igényelt változások szerint.
- Két legismertebb modell a támogatására:
 - **Inkrementális fejlesztés:**
 - a szoftverspecifikáció, a tervezés, az implementálás, kis inkrementációs lépésekre van felosztva.
 - **Spirális fejlesztés:**
 - a rendszer fejlesztése egy belülről kifelé tartó spirálvonalat követ
- Az iteratív folyamat lényege: a specifikáció a szoftverrel összekapcsolva készül.

32

Inkrementális fejlesztés

- **Egy köztes megközelítés a vízesésmodell és az evolúciós fejlesztési modellek között.**

- **A vízesésmodell előnye:** egyszerűen menedzselhető, mert külön választja az egyes fázisokat.
 - Hátrány: robusztus rendszerek jöhetnek létre, amik esetleg alkalmatlanok a változtatásokra.
- **Az evolúciós megközelítés:** megengedettek a követelményekkel és tervezésekkel kapcsolatos döntések elhagyása.
 - Gyengén strukturált és nehezen megérthető rendszerekhez vezethetnek.

33

Inkrementális fejlesztés lépései

- **1. A megrendelő meghatározza:**
 - nagy körvonalakban a rendszer által nyújtandó szolgáltatásokat,
 - mely szolgáltatások fontosabbak, melyek kevésbé.
- **2. A követelmények inkremensekben való megfogalmazása és hozzárendelése:**
 - függ a szolgáltatás prioritásától,
 - a magasabb prioritású szolgáltatásokat hamarabb kell biztosítani a megrendelő felé.

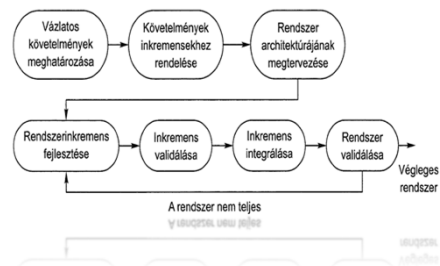
34

Inkrementális fejlesztés lépései

- **3. Az inkremensek által előállítandó szolgáltatások követelményeit részletesen definiálni kell.**
- **4. Az inkremensek kifejlesztése.**
 - Sor kerülhet további követelmények elemzésére, de az adott lépés követelményei nem módosíthatók.
- **5. Az elkészült új inkremensek integrálása a már kész inkremensekkel.**
 - A rendszerfunkciók köre így egyre bővül.
- **Ha egy inkremens elkészült, a rendszer bizonyos funkcióit akár be is üzemeltethetik.**
 - Cél: tapasztalat szerzés a rendszerrel kapcsolatban.

35

A fejlesztési modell



36

Áttekintés

- **Előnyök:**

- A megrendelőnek nem kell megvárnia míg a teljes rendszer elkészül, a szoftver már menet közben használhatóvá válik.
- A megrendelők használhatják a korábbi inkremenseket mint prototípusokat, ami által tapasztalatokat szerezhetnek.
- Kisebb a kockázata annak, hogy a teljes projekt kudarcba fullad.
- A magasabb prioritású inkremenseket szállítjuk le hamarabb, ezért mindig a legfontosabb szolgáltatások lesznek többet tesztelve.
 - kisebb a hiba esélye a rendszer legfontosabb részeiben.

37

Áttekintés

- **Hátrányok:**

- Az inkremenseknek megfelelően kis méretűeknek kell lenni.
- minden inkrementációs lépésnek szolgáltatni kell valami rendszerfunkciót.
- nehézkessé válhat a megrendelő követelményeit megfelelő méretű inkrementációs lépésekre bontani.

38

SPIRÁLIS FEJLESZTÉS...

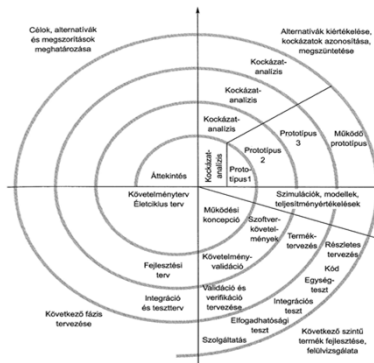
39

Spirális fejlesztés

- Boehm javasolta először már 1988-ban
 - azóta széles körben elterjedt az irodalomban és a gyakorlatban.
- A szoftverfolyamatot nem tevékenységek és közöttük található esetleg visszalépések sorozataként tekinti, hanem inkább egy spirálként reprezentálja.
- A spirál minden egyes körben a szoftverfolyamat egy-egy fázisát reprezentálja.

40

A spirális modell



41

A spirál négy szektora

- **1. Célok kijelölése:** Az adott projektfázis által kitűzött célok meghatározása
 - A folyamat megszorításainak azonosítása,
 - A kapcsolódó menedzselési terv vázolása.
 - A projekt kockázati tényezőinek felismerése, és stratégiák tervezése.
- **2. Kockázat becslése:** Minden kockázati tényező esetén részletes elemzésre kerül sor.
 - Lépéseket kell tenni a kockázat csökkentése, megszüntetése érdekében.

42

A spirál négy szektora

- **3. Fejlesztés és validálás:** a kiértékelés után egy fejlesztési modellt kell választani a problémának megfelelően. Pl. evolúciós, vízésés, stb modellek.
 - Tervezés, fejlesztés, tesztelés, validálás.
- **4. Tervezés:** a folyamat azon fázisa, ahol dönteni kell, hogy folytatódjon-e egy következő ciklussal, vagy sem.
 - Folytatás esetén vázolni kell a következő fázist.
 - Fejlesztési terv, integrációs tesztterv.

43

Áttekintés

- Miben más a spirális fejlesztési modell az egyéb szoftverfolyamat-modelltől?
- A modell expliciten számol a kockázati tényezőkkel, amelyek problémákat okozhatnak a projektben.
 - Pl.: a határidő- és költségűllépések.

44

Köszönöm a figyelmet!

45