

Dr. Mileff Péter

SZOFTVERFEJLESZTÉS

1. ELŐADÁS

1

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 működjek.

2

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.

3

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.
- **Cél:** úgy kell kialakítani, hogy kiaknázzák a szervezetben belül az emberek képességeit és a fejlesztő rendszer jellegzetességeit.

4

A folyamat közös fázisai

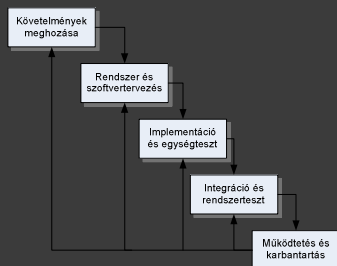
- **Szoftverspecifikáció:** A szoftver funkcióit, illetve annak megszorításait definiálni kell.
- **Szoftvertervezés és implementáció:** A specifikációnak megfelelő szoftvert elő kell állítani.
- **Szoftvervalidáció:** a szoftvert validálni kell, hogy biztosítsuk, azt fejlesztettük, amit az ügyfél kíván.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

A szoftverfolyamat modelljei

- **A szoftverfolyamat absztrakt reprezentációja:**
 - speciális perspektívából reprezentál egy folyamatot
 - Részleges információk a folyamatról, mert általánosak.
- Ismertebb modellek:
 - **Vízesésmmodell:** Ez a folyamat alapvető tevékenységeit a folyamat különálló fázisainak tekintti.
 - **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.

A vízesésmmodell

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



1 fázis: követelmények elemzése és meghatározása

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

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 közöttük lévő kapcsolatok tervezése és leírása.

9

3 fázis: implementáció és egységteszt

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

10

4 fázis: integráció és rendszerteszt

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

11

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.

12

Áttekintés

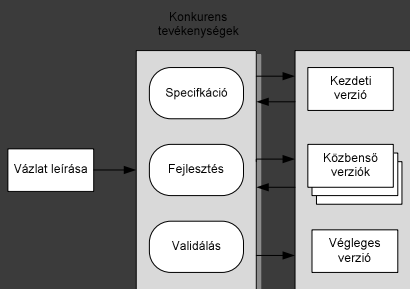
- A fázisok eredménye egy dokumentum.
- Egy fázis csak akkor indulhat, ha az előző befejeződött.
- A folyamat **nem egyszerű lineáris modell**, hanem a fejlesztési tevékenységek iterációjának sorozata.
 - Az iterációk költségesek, így gyakran befagyasztyák őket. A problémák megoldása később, vagy soha.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

EVOLÚCIÓS FEJLESZTÉS...

Evolúciós fejlesztés

- **Alapötlete:**
 - a fejlesztőcsapat kifejleszt egy kezdeti implementációt,
 - majd azt a felhasználókkal véleményezteteti,
 - 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.

Evolúciós modell



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.

17

Áttekintés

Előnye:

- **hatékonyabb** a vízesésmodellnél, 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.

18

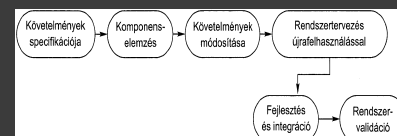
KOMPONENS ALAPÚ FEJLESZTÉS...

19

Komponens alapú fejlesztés

Alapgondolata az újrafelhasználható komponensekből való építkezés.

- A szoftverfolyamatokban megtalálhatók a komponensek újrafelhasználása:
 - Korábbi kód átdolgozása, felhasználása, általánosítása.



20

Komponenselemzés

- A követelményspecifikáció alapján **komponensek keresése**, hogy melyek implementálják azokat.
- Általában nincs egzakt illeszkedés, a felhasznált komponens a funkciók csak egy részét nyújtja.

21

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 és alternatív megoldást kell keresni.

22

Rendszertervezés újrafelhasználással

- Ez a szakasz felelős a rendszer szerkezetének **tervezéséért**:
 - Számba kell venni, hogy milyen komponenseket akarnak újrafelhasználni, és úgy alakítani a szerkezetet, hogy azok működhessenek.
 - Ha nincs elérhető újrafelhasználható komponens, akkor új szoftverek is kifejleszthetők.

23

Fejlesztés és integráció

- A nem megvásárolt komponenseket ki kell fejleszteni és a rendszerbe integrálni.
- A rendszer-integráció ebben a modellben sokkal inkább a fejlesztési folyamat része, mint különálló tevékenység.

24

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.

25

FOLYAMAT-ITERÁCIÓ

1. INKREMENTÁLIS FEJLESZTÉS
2. SPIRÁLIS FEJLESZTÉS

26

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.

27

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 megközelítésnél pedig megengedettek a követelményekkel és tervezésekkel kapcsolatos döntések elhagyása.
 - Gyengén strukturált és nehezen megérthető rendszerekhez vezethetnek.

28

Inkrementális fejlesztés lépései

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

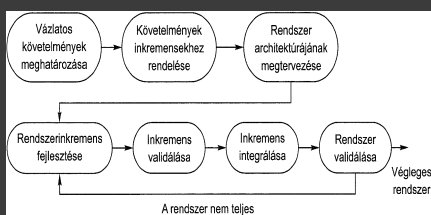
29

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:** tapasztalatszerzés a rendszerrel kapcsolatban.

30

A fejlesztési modell



31

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

32

Áttekintés

● **Hátrányok:**

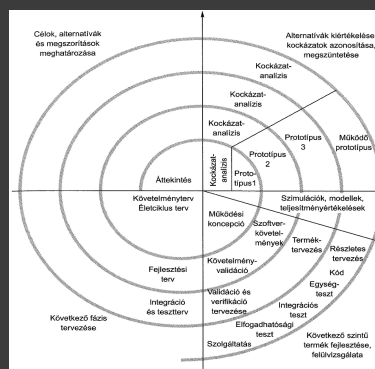
- Az inkrementeknek megfelelően kis méretűeknek kell lenni.
- minden inkrementációs lépésnek szolgáltatni kell valami rendszerfunkciót.
- nehézkessé válhat a megrendelő követelményeit megfelelő méretű inkrementációs lépésekre bontani.

SPIRÁLIS FEJLESZTÉS...

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.

A spirális modell



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 érdekében.

37

A spirál négy szektora

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

38

Áttekintés

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

39

Köszönöm a figyelmet!

40