

Szoftvertechnológia

1. előadás

A szoftverfejlesztési folyamat

Giachetta Roberto

groberto@inf.elte.hu
http://people.inf.elte.hu/groberto

„In short, software is eating the world.”

(Marc Andreessen)

A szoftverfejlesztési folyamat

Szoftverfejlesztés

- A szoftverek nélkülözhetetlen alkotóelemei a modern világnak
 - számos célt szolgálhatnak
 - különböző felépítésűek, működési elvűek
 - megvalósításuk módja jelentősen eltérhet
- A szoftverekben sok hiba található, a szoftverfejlesztési munkák nagyrésze kudarcba fullad, ennek okai:
 - egyre nagyobb számban, egyre összetettebb szoftverekre van szükség
 - alacsonyak az elvárások a szoftverekkel szemben
- Nagy szükség van a *professzionális szoftverfejlesztésre*

A szoftverfejlesztési folyamat

Jelentős szoftverhibák

- *Intel*, Pentium processzor (1994): hibás lebegőpontos számítás
- *ESA*, Ariane-5 (1996): adat túlsordulás
- *NASA*, Mars Climate Orbiter (1998): mértékegység tévesztés
- *General Electric*, észak-amerikai áramkimaradás (2003): kiéheztetés
- *World of Warcraft*, Corrupted Blood Incident (2005)
- *OpenSSL*, Heartbleed (2012): adatszivárgás
- *Apple*, goto fájl (2014): hibás elágazás kezelés
- *Toyota*, ETCS gyorsulás szabályozás (2014): verem túlsordulás
- *Steam*, felhasználói fiók törlése (2015): útvonal ellenőrzés hiánya
- *Boeing*, 787 Dreamliner (2015): adat túlsordulás

A szoftverfejlesztési folyamat

Minőségi mutatók

- A szoftvereknek megfelelő színvonalon kell biztosítani az elvárt funkciókat, amit a szoftver *minőségi mutatóival* (*quality characteristics*) írhatunk le
 - *karbantarthatóság* (*maintainability*): módosíthatóság, továbbfejleszthetőség lehetőségei
 - *megbízhatóság és biztonság* (*dependability and security*): meghibásodások valószínűsége, támadásokkal szembeni védelem, sebezhetőségi pontok
 - *hatékonyság* (*efficiency*): erőforrások használata, korlátai, válaszidő, skálázhatóság
 - *használhatóság* (*acceptability*): érthetőség, használat elsajátítása, ergonómia

A szoftverfejlesztési folyamat

A szoftvertechnológia

- Egy szoftvernek, mint terméknek gyártási technológiára van szüksége, amely garantálja a program funkcióit, minőségét, költségét és határidejét
- *A szoftvertechnológia feladata szoftverek rendszerezett, felügyelt, minősített fejlesztése, működtetése és karbantartása*
 - *a szoftver a program(ok), dokumentáció(k), konfiguráció(k), valamint adatok együttese*
- A szoftverek többsége nagy méretű, nagy bonyolultságú programrendszer, amely
 - rendszerint csapatmunkában készül
 - hosszú élettartamú, karbantartást és bővítést igényel

A szoftverfejlesztési folyamat

Szoftvertechnológiai projekt

- A szoftver fejlesztésének folyamatát *projekt*nek, előállításának felügyeletét *projektmenedzselés*nek nevezzük
- A projektért felelős személy a *projektmenedzser (project manager)*, aki
 - biztosítja, hogy a szoftver megfelel az előírt minőségnek, és elkészül a megadott határidőre a költségkereten belül
 - szervezi, irányítja, ütemezi a projektben részt vevő csapat munkáját, és biztosítja a szükséges hardver és szoftver erőforrásokat
 - garantálja a módszerek és szabványok alkalmazását
 - gondoskodik a projekt dokumentáltságáról

ELTE IK, Szoftvertechnológia

1:7

A szoftverfejlesztési folyamat

Szoftvertechnológiai projekt

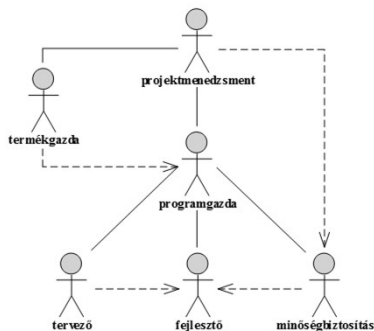
- A szoftverfejlesztési csapatnak számos további tagja lehet, akik különböző szerepeket töltenek be, pl.:
 - *termékgazda (product management)*: üzleti folyamatok, prioritások és elfogadási feltételek kezelése
 - *programgazda (program management)*: fejlesztés ütemezése, feladatok elosztása és követése
 - *tervező (architect)*: szoftver magas szintű tervének elkészítése, technikai döntések kezelése
 - *fejlesztő (developer)*: szoftver implementációja
 - *minőségbiztosítás (quality assurance)*: tesztelés tervezése, magvalósítása, minőségi kritériumok ellenőrzése

ELTE IK, Szoftvertechnológia

1:8

A szoftverfejlesztési folyamat

Szoftvertechnológiai projekt



ELTE IK, Szoftvertechnológia

1:9

A szoftverfejlesztési folyamat

A szoftver életciklus

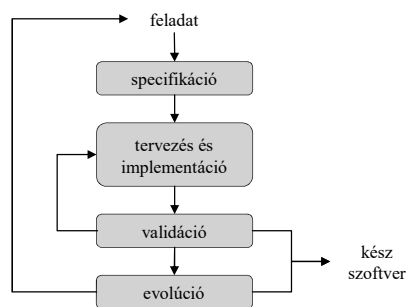
- Minden szoftver rendelkezik *életciklussal*, amely meghatározza létét a feladat kitűzésétől a program használatának befejeztéig
- Az életciklus általában négy fő fázisra bontható:
 1. *specifikáció*: a szoftver funkcionalitásának és megszorításainak megadása
 2. *tervezés és implementáció*: a specifikációnak megfelelő szoftver előállítása
 3. *verifikáció és validáció*: a szoftver ellenőrzése a specifikációnak történő megfelelésre
 4. *evolúció*: a szoftver továbbfejlesztése a változó elvárásoknak megfelelően

ELTE IK, Szoftvertechnológia

1:10

A szoftverfejlesztési folyamat

A szoftver életciklus



ELTE IK, Szoftvertechnológia

1:11

A szoftverfejlesztési folyamat

Specifikáció

- A *specifikáció (software specification)* célja a feladatot megoldó szoftver funkcióinak tisztázása, a rendszerre és a fejlesztésre vonatkozó elvárások megadása
 - feltérképezi a követelményeket felhasználói, valamint fejlesztői szempontból, lépései:
 - megvalósíthatósági elemzés
 - követelmény feltárás és elemzés
 - követelmény specifikáció
 - követelmény validáció
 - eredménye a *szoftver követelmény-leírása (software requirements specification)*

ELTE IK, Szoftvertechnológia

1:12

A szoftverfejlesztési folyamat

Tervezés és implementáció

- A *szoftver tervezése és implementációja (software design and implementation)* feladata a specifikáció átalakítása egy végrehajtható rendszerre
 - meghatározza a rendszer szerkezetét (felépítés), felületét (be- és kimenet), működését (alkalmazott algoritmusok, kommunikációs folyamatok)
- a folyamat során elkészül a *szoftver rendszerterve (software design description)*, amely tartalmazza a program statikus és dinamikus szerkezetét, a kommunikációs csatornák feltérképezését, az implementációs és tesztelési tervet
- elkészíthető a *szoftver prototípusa (prototype)*, amely a program egyszerűsített megvalósítását tartalmazza

ELTE IK, Szoftvertechnológia 1:13

A szoftverfejlesztési folyamat

Tervezés és implementáció

- az implementációhoz megfelelő *szoftverfejlesztési környezetet* kell használnunk, a programkód változásait *verziókövetéssel* tartjuk nyilván
- az implementáció részeként az egyes programegységek tesztelése is megtörténhet
- a szoftverek tervezésének és programozásának módszerét nevezzük *programozási paradigmának*
 - meghatározza a programozási stílust, az absztrakciós szintet
 - meghatározza az alkalmazható programozási nyelvek körét is, és fordítva

ELTE IK, Szoftvertechnológia 1:14

A szoftverfejlesztési folyamat

Programozási paradigmák

The diagram illustrates programming paradigms. It is divided into 'deklaratív' (declarative) and 'imperatív' (imperative) sections. In the declarative section, 'csak adatstruktúrák' (only data structures) leads to 'Turing-teljes' (Turing-complete), which includes 'λ-kalkulus' (lambda calculus) and 'funkcionális' (functional) programming. 'funkcionális' leads to 'ekvivalencia' (equivalence) and 'relációs (logikai)' (relational/logical). 'funkcionális' also leads to 'adatfolyam' (data flow), which leads to 'folyam-alapú' (flow-based) programming. In the imperative section, 'tartományleíró rekord' (domain description record) leads to 'eljárás' (procedure), which includes 'állapot' (state) and 'procedurális' (procedural) programming. 'procedurális' leads to 'lokalizáció' (localization) and 'strukturált' (structured) programming. 'strukturált' leads to 'öröklődés' (inheritance) and 'objektum-orientált' (object-oriented) programming.

ELTE IK, Szoftvertechnológia 1:15

A szoftverfejlesztési folyamat

Validáció és evolúció

- A *verifikáció és validáció (software verification and validation)* célja megmutatni, hogy a rendszer megfelel a specifikációnak, valamint a felhasználói elvárásoknak
 - alapvetően tesztelés, amely több fázisban, több módszerrel történik (a felhasználói tesztek csak az utolsó lépésben történnek)
- Az *evolúció (software evolution)* során új követelményeknek megfelelően bővítjük a szoftvert, illetve korrigáljuk a felmerülő hibákat
 - átlagosan a szoftver élettartamának 80%-a, ezért eleve bővíthetőre, módosíthatóra kell kialakítani a szoftvert

ELTE IK, Szoftvertechnológia 1:16

A szoftverfejlesztési folyamat

A szoftver életciklus

- További lépések is kísérhetik a fejlesztési folyamatot, pl.
 - kihelyezés (deployment)*: a program üzembe állítása, és elérhetővé tétele
 - tréning és támogatás (training and support)*: a felhasználókkal való kapcsolattartás (annak biztosítása, hogy a szoftvert megfelelően tudják kezelni és használni)
- A szoftver dokumentációja két részből tevődik össze:
 - felhasználói dokumentáció*, amely tartalmazza a szoftver üzembe helyezésének, funkcióinak bemutatását
 - fejlesztői dokumentáció*, amely tartalmazza a szoftver megvalósítását folyamatát és részletes ismertetését

ELTE IK, Szoftvertechnológia 1:17

A szoftverfejlesztési folyamat

Ütemterv

- A szoftver életciklus fázisai (*feladatai*) további fázisokra (*részfeladatokra*) tagolhatóak, így egy hierarchikus feladatszerkezetet kapunk
 - az egyes feladatokra erőforrásokat és időkorlátot adhatunk
 - az egyes feladatok között *függőségeket* állapíthatunk meg (a feladat nem kezdhető el, amíg a függősége el nem készül)
 - ezek alapján elkészíthetjük a *projekt ütemtervét*
 - tartalmazza a feladatok időbeli beosztását, függőségeit, felelőseit, így áttekinthetővé teheti az erőforrás szükségleteket
 - általában a specifikáció során készül el, de később módosulhat

ELTE IK, Szoftvertechnológia 1:18

A szoftverfejlesztési folyamat

Mérföldkövek

- A feladatokhoz *mérföldköveket (milestone)* rendelhetünk, amelyek lehetőséget adnak a projekt haladásában történő betekintésre
 - a mérföldkö egy adott cél adott időpontra történő elérését jelenti, így *névvel, eseménnyel, céllal* rendelkezik
 - a mérföldkövek be nem tartása általában korrekciókat követel a projekt lefutásában
 - kellően konkrétak, ellenőrizhetőek, számon kérhetőek kell lenni (akár a termékigazda számára is)
 - a fő mérföldkövek az egyes fázisok lezárását jelentik, ezen kívül számos további mérföldkö adható

ELTE IK, Szoftvertechnológia 1:19

A szoftverfejlesztési folyamat

Ütemterv

The Gantt chart illustrates a project schedule with tasks F1 through F7 and milestones M1, M2, and M3. The x-axis represents time points from t_1 to t_{10} . Task F1 starts at t_1 and ends at t_2 . F2 starts at t_2 and ends at t_4 . F3 starts at t_3 and ends at t_5 . F4 starts at t_4 and ends at t_6 . F5 starts at t_5 and ends at t_8 . F6 starts at t_7 and ends at t_9 . F7 starts at t_8 and ends at t_{10} . Milestones M1, M2, and M3 are marked at t_3 , t_7 , and t_{10} respectively. Arrows indicate dependencies between tasks.

ELTE IK, Szoftvertechnológia 1:20

A szoftverfejlesztési folyamat

Az UML

- A szoftverfejlesztési életciklust folyamatosan követi a modellezés, ennek eszköze az *egységes modellezési nyelv (Unified Modeling Language, UML)*, amely egy öt pillérű szemléletrendszerrel rendelkezik:
 - használati*: a szoftver szolgáltatásai és azok kapcsolata a felhasználókkal
 - szerkezeti* (statikus): a rendszer és a programegységek felépítése, kapcsolatai
 - dinamikus*: a programegységek viselkedése
 - implementációs*: a megvalósítás szempontjai, komponensei
 - környezeti*: hardver és szoftver erőforrások

ELTE IK, Szoftvertechnológia 1:21

A szoftverfejlesztési folyamat

Az UML

The diagram shows a central oval labeled 'használati' (usage) surrounded by four quadrants: 'szerkezeti' (structural), 'dinamikus' (dynamic), 'implementációs' (implementation), and 'környezeti' (environmental).

ELTE IK, Szoftvertechnológia 1:22

A szoftverfejlesztési folyamat

Szoftverfejlesztési modellek

- Amellett, hogy a szoftverfejlesztés betartja az életciklus fázisait, a folyamat lefolyása különféle módokon történhet, amiket *szoftverfejlesztési módszereknek* nevezünk
 - klasszikus módszerek: *vizesés, prototipizálás, inkrementális, iteratív, spirális, V-Model*
 - agilis módszerek: *Scrum, Lean, Kanban, XP* (extreme programming), *RAD* (rapid application development)
 - speciális célú módszerek: *BDD* (behavior-driven development), *TDD* (test-driven development), *FDD* (feature-driven development)
 - formális módszerek: *B-módszer*

ELTE IK, Szoftvertechnológia 1:23

A szoftverfejlesztési folyamat

Szoftvereszközök

- A fejlesztőcsapat munkáját megfelelő szoftvereszközökkel kell alátámasztani
 - projektirányítási eszközzel (project tracking system)*, amely támogatja a dokumentálást és a feladatok követését
 - fejlett *tervezőeszközzel (case tool)*, ahol a fejlesztés folyamata és a felelőség is nyomon követhető
 - integrált fejlesztőkörnyezettel (IDE)*, amely elősegíti a csapatmunkát
 - verziókövető rendszerrel (revision control system)*, amely lehetővé teszi a programkód változásainak követését
 - folyamatos integrációs eszközzel (continuous integration)*, amely elősegíti a tevékenységek automatizálását

ELTE IK, Szoftvertechnológia 1:24

A szoftverfejlesztési folyamat Projektirányítási eszközök lehetőségei	
<ul style="list-style-type: none"> • A <i>projektirányítási eszköz</i> lehetőséget ad a projekt menedzselésének ellátására <ul style="list-style-type: none"> • általában webes felületű eszköz, amely bárholonnan elérhető és használható • főbb funkciói: <ul style="list-style-type: none"> • fejlesztés ütemtervének, kockázatainak rögzítése • egyszerű és folyamatos dokumentálás lehetősége • feladatok, tevékenységek, hibák rögzítése, és a kapcsolatos tevékenységek követése • integrált forráskód böngészés, és forrástörténet áttekintés (verziókezelés) • pl.: <i>Trac, Redmine, Team Foundation Server (TFS)</i> 	1:25
ELTE IK, Szoftvertechnológia	

A szoftverfejlesztési folyamat Projektirányítási eszközök lehetőségei	
<ul style="list-style-type: none"> • A rendszerek lehetőséget adnak a fejlesztők számára feladatok kitűzésére, valamint a tesztelők számára a programban fellelhető hibák jelzésére • a feladatokat úgynevezett <i>cédulák (ticket, issue)</i> segítségével írhatóak ki <ul style="list-style-type: none"> • jelölhetnek új funkcionalitást (<i>feature</i>), hibát (<i>bug</i>), egyéb fejlesztési feladatot (<i>task</i>), vagy dokumentációs feladatot (<i>documentation</i>) • megadható a leírása, felelőse, határideje • kommentálhatóak, lezárhatóak, újra kinyithatóak • a cédulák biztosítják a fejlesztési és tesztelési folyamat naplózását 	1:26
ELTE IK, Szoftvertechnológia	

A szoftverfejlesztési folyamat Projektvezető szolgáltatások	
<ul style="list-style-type: none"> • A <i>projektvezető szolgáltatások (project hosting services)</i> olyan online szolgáltatások, amelyek a projekttel kapcsolatos legtöbb funkcionalitást integrálják <ul style="list-style-type: none"> • projektmenedzsment, kód tárolás, verziókövetés, dokumentáció, folyamatos integráció, kihelyezés • integrálhatóak projektirányítási eszközökkel, fejlesztőeszközzel • garantálják a kód épségét, a folyamatos rendelkezésre állást • általában nyílt forráskódú szoftverek esetén ingyenes a használatuk • pl.: <i>SourceForge, CodePlex, GitHub, GitLab</i> 	1:27
ELTE IK, Szoftvertechnológia	

Esettanulmányok Tic-Tac-Toe játék	
<p><i>Feladat:</i> Készítsünk egy Tic-Tac-Toe programot, amelyben két játékos küzdhet egymás ellen.</p> <ul style="list-style-type: none"> • a programban jelenjen meg egy játéktábla, amelyen végig követjük a játék állását (a két játékos az 'X' és 'O' jelekkel ábrázoljuk) • legyen lehetőség a játékosok neveinek megadására, új játék indítására, valamint játékban történő lépésre (felváltva) • a program kövesse végig, melyik játékos hány kört nyert • program automatikusan jelezzen, ha vége egy játéknak, és jelenítse meg a játékosok pontszámait 	1:28
ELTE IK, Szoftvertechnológia	

Esettanulmányok Marika néni kávézója	
<p><i>Feladat:</i> Készítsük el Marika néni kávézójának eladási nyilvántartását végigkövető programot.</p> <ul style="list-style-type: none"> • a kávézóban 3féle étel (hamburger, ufó, palacsinta), illetve 3féle ital (tea, narancslé, kóla) közül lehet választani • az ételek ezen belül különfélék lehetnek, amelyre egyenként lehet árat szabni, és elnevezni, az italok nevei és árai rögzítettek • a program kezelje a rendeléseket, amelyekben tetszőleges tételek szerepelhetnek, illetve a rendelés kapcsolódhat egy törzsvásárlóhoz • biztosítsunk lehetőséget a függőben lévő rendeléseket lekérdezésére, valamint napi, havi és törzsvásárlói számra összesített nettó/bruttó fogyasztási statisztikák követésére 	1:29
ELTE IK, Szoftvertechnológia	

Esettanulmányok Memory játék	
<p><i>Feladat:</i> Készítsünk egy <i>Memory</i> kártyajátékot, amelyben két játékos küzd egymás ellen, és a cél kártyapárok megtalálása a játéktáblán.</p> <ul style="list-style-type: none"> • a játékosok felváltva lépnek, minden lépésben felfordíthatnak két kártyát • amennyiben a kártyák egyeznek, úgy felfordítva maradnak és a játékos ismét léphet, különben visszafordulnak, és a másik játékos következik • a játékot az nyeri, aki több kártyapárt talált meg • lehessen a játékosok neveit megadni, kártyacsomagot választani, valamint a kártyák számát (a játéktábla méretét) szabályozni 	1:30
ELTE IK, Szoftvertechnológia	

Esettanulmányok

Utazási ügynökség

Feladat: Készítsük el egy utazási ügynökség apartmanokkal foglalkozó rendszerét.

- az apartmanok épületekben találhatóak, amelyek városokban helyezkednek el
- az épületek különböző adatokkal (leírás, szolgáltatások, pontos hely, tengerpart távolság, ...), valamint képekkel rendelkeznek
- a vendégek számára biztosítsunk egy webes felületet, amelyen keresztül apartmanokat kereshetnek, foglalhatnak
- a munkatársak számára biztosítsunk egy alkalmazást, amelyben szerkeszthetik az apartmanok adatait, képeit, valamint kezelhetik a foglalásokat