

## Térinformatikai és távérzékelési alkalmazások fejlesztése

### A szoftverfejlesztés technikai támogatása

© 2016 Giachetta Roberto  
groberto@inf.elte.hu  
<http://people.inf.elte.hu/groberto>

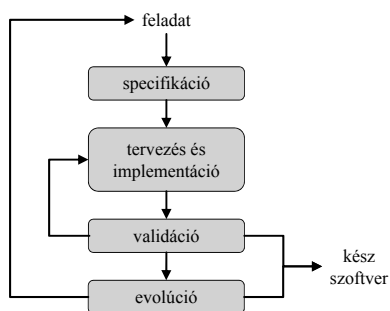
#### A szoftverfejlesztés technikai támogatása

##### 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 előállítás
  3. *verifikáció és validáció*: a szoftver ellenőrzése a specifikációnak történő megfelelésre
  4. *evolúció*: a szoftver továbbfejlesztése a változó elvárásoknak megfelelően

#### A szoftverfejlesztés technikai támogatása

##### A szoftver életciklus



#### A szoftverfejlesztés technikai támogatása

##### Szoftvereszközök

- A fejlesztőcsapat munkáját megfelelő szoftvereszközökkel kell alátámasztani
  - *projektmenedzsment eszközzel (project tracking system)*, amely támogatja a dokumentálást és a feladatok követését
  - *fejlett tervezőeszközzel (case tool)*, ahol a fejlesztés folyamata és a felelősség is nyomon követhető
  - *integrált fejlesztőkörnyezettel (IDE)*
  - *verziókövető rendszerrel (revision control system)*, amely lehetővé teszi a programkód változásainak követését
  - *folytonos integrációs (continuous integration) rendszerrel*, amely biztosítja a hibák korai kiszűrését

#### A szoftverfejlesztés technikai támogatása

##### Projektmenedzsment eszközök lehetőségei

- A projektmenedzsment eszköz lehetőséget ad az alábbiakra:
  - fejlesztés ütemtervének, kockázatainak meghatározása
  - fejlesztés egyszerű és folyamatos dokumentálásának lehetősége és generálása
  - feladatok, tevékenységek rögzítése, követése
  - a tesztelés során előfordult hibák rögzítése, a javítási folyamat követése
  - integrált verziókezelés és forráskód böngészés
  - webes vagy grafikus felület, amely biztosítja a könnyű használatot, és bárholnan való elérést

#### A szoftverfejlesztés technikai támogatása

##### Ütemterv és időzítés

- A szoftver lehetőséget ad, hogy a projekt ütemtervét elkészítsük, és azt folyamatosan szem előtt tarthassuk
  - definiálhatunk mérföldköveket, amelyre adott feladatokat el kell végezni
  - a fejlesztők külön-külön láthatják a saját feladataikat, menedzselhetik annak előrehaladását
  - beoszthatjuk a fejlesztési lépések erőforrásait
  - definiálhatunk függőségeket a programrészek között
  - kezelhetjük az egyes fejlesztési lépések időbeli lefolyását, előrevetíthetjük a tervezettől való eltérések hatásait az erőforrásokra, illetve a további fejlesztési időkre

## A szoftverfejlesztés technikai támogatása

### Feladat és hibakövetés

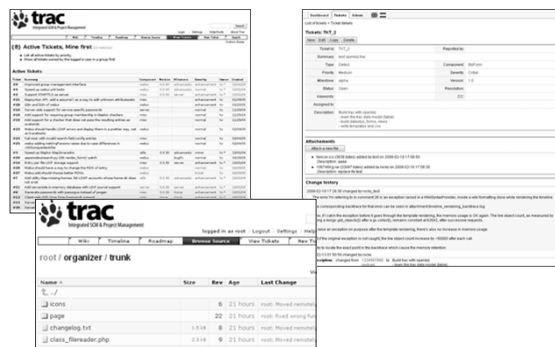
- A rendszerek lehetőséget adnak a tervezők számára feladatok kitűzésére, valamint a tesztelők számára a programban fellelhető hibák jelzésére
  - a feladatokat úgynevezett *cédulák* (*ticket, issue*) segítségével írhatóak ki
    - jelölhetnek új funkcionalitást (*feature*), hibát (*bug*), egyéb fejlesztési feladatot (*task*), vagy dokumentációs feladatot (*documentation*)
    - megadható a leírása, felelőse, határideje
    - kommentálhatóak, lezárhatóak, újra megnyithatóak
  - a cédulák biztosítják a fejlesztési és tesztelési folyamat naplózását

ELTE IK, Térinformatikai és távérzékelési alkalmazások fejlesztése

7

## A szoftverfejlesztés technikai támogatása

### A Trac projektmenedzser



ELTE IK, Térinformatikai és távérzékelési alkalmazások fejlesztése

8

## A szoftverfejlesztés technikai támogatása

### A Redmine projektmenedzser



ELTE IK, Térinformatikai és távérzékelési alkalmazások fejlesztése

9

## A szoftverfejlesztés technikai támogatása

### Projektmenedzsmet eszközök

- Az eszközök felületi része alkalmas webes technológiával, míg az adattárolás adatbázis-motor segítségével valósítják meg
  - a legtöbb eszköz szabad forráskódú, és a projektvezetés ugyanazon eszközzel van menedzselve
- Néhány népszerű projektmenedzser:
  - Trac*: Python alapú, MySQL/SQLite/PostgreSQL adatbázis háttérrel
  - Redmine*: Ruby on Rails alapú, MySQL/SQLite/PostgreSQL adatbázis háttérrel
  - Microsoft Team Foundation Server*: ASP.NET, Exchange és MSSQL alapú, elsősorban Visual Studio számára

ELTE IK, Térinformatikai és távérzékelési alkalmazások fejlesztése

10

## A szoftverfejlesztés technikai támogatása

### Projektvezető szolgáltatások

- A *projektvezető szolgáltatások* (*project hosting services*) általában rendelkezésre bocsátanak több projektfejlesztő eszközt
  - projektmenedzsmet, kód tárolás, kód megtekintés, verziókövetés, dokumentáció (Wiki), levelezési lista, adatbázis hozzáférés
  - általában nyílt forráskódú szoftverek esetén ingyenes a szolgáltatás
  - pl.: *SourceForge*, *GitHub*, *CodePlex*
  - egyes szolgáltatások bizonyos programozási nyelvek, vagy témakör köré csoportosulnak (pl. *mozdev*, *RubyForge*)

ELTE IK, Térinformatikai és távérzékelési alkalmazások fejlesztése

11

## A szoftverfejlesztés technikai támogatása

### Az integrált fejlesztőkörnyezet

- Az implementációhoz megfelelő *integrált fejlesztőkörnyezet* (*IDE*) szükséges (pl. *Eclipse*, *Visual Studio*, *Xcode*, *NetBeans*)
  - nem csak az implementáció folyamatát, de a teljes szoftver életciklust (tervezés, tesztelés) támogatja
  - a fejlesztést kód-kiemeléssel (*syntax highlight*), kód-kiegészítésekkel (*code-snippet*, *intelligent code completion*), kódstílus követéssel, illetve kód-újratervezési eszközökkel támogatja
  - a tesztelést nyomkövetéssel (*debugging*), egységtesztek (*unit test*), illetve teljesítményteszteléssel támogatja
  - kapcsolatot biztosít (vagy integrálja) a verziókövetéshez és a projektmenedzsmet eszközhöz

ELTE IK, Térinformatikai és távérzékelési alkalmazások fejlesztése

12

### A szoftverfejlesztés technikai támogatása

#### Csapatban történő implementáció

- A szoftverek általában csapatban készülnek
  - az implementáció egy egységes kódolási stílus mentén történik, egységes eszköztárral
  - minden fejlesztő csak a saját kódján dolgozik
    - verziókövetés esetén általában külön fejlesztési ágban tevékenykedik
    - amennyiben más kódjában hibát talál, hibajelzést tesz
    - az általa biztosított interfészeket csak egyeztetés után módosítja
    - dokumentálja (kommentezi), illetve teszteli a saját kódját (elkészíti a megfelelő egységtesztet)

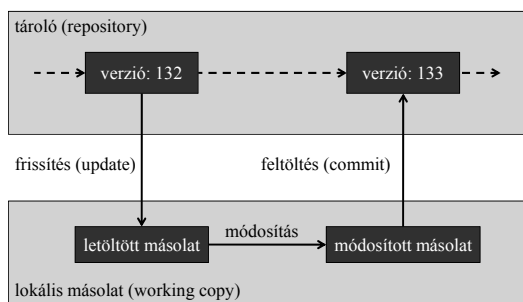
### A szoftverfejlesztés technikai támogatása

#### Verziókövető rendszerek

- Mivel az implementáció több lépésben, és sokszor párhuzamosan zajlik, szükséges, hogy az egyes programállapotok, jól követhetőek legyenek, ezt a feladatot a *verziókövető rendszerek (revision control system)* látják el
  - pl. *CVS, Apache Subversion (SVN), Mercurial*
  - egy közös tárolóban (*repository*) tartják kódokat
  - ezt a fejlesztők lemásolják egy helyi munkakönyvtárba, és amelyben dolgoznak (*working copy*)
  - a módosításokat visszautlítik a központi tárolóba (*commit*)
  - a munkakönyvtárakat az első létrehozás (*checkout*) után folyamatosan frissíteni kell (*update*)

### A szoftverfejlesztés technikai támogatása

#### Verziókövető rendszerek



### A szoftverfejlesztés technikai támogatása

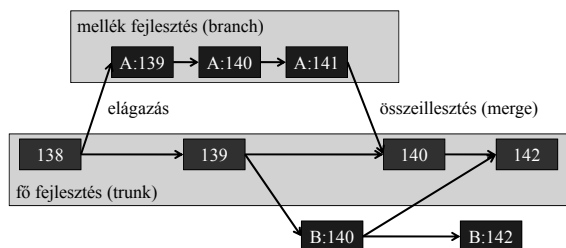
#### Verziókövető rendszerek

- A rendszerek lehetővé teszik:
  - az összes eddig változat (*revision*) eltárolását, illetve annak letöltési lehetőségét
  - a fő fejlesztési vonal (*baseline, master* vagy *trunk*) és a legfrissebb változat (*head*) elérését, új változat feltöltését annak dokumentálásával
  - az egyes változatok közötti különbségek nyilvántartását fájlanként és tartalmanként (akár karakterek szintjén)
  - változtatások visszavonását, korábbi változatra visszatérést
  - konfliktust okozó módosítások ellenőrzését, illetve megoldását (*resolve*)

### A szoftverfejlesztés technikai támogatása

#### Verziókövető rendszerek

- a folyamat elágazását, és ezáltal újabb fejlesztési folyamatok létrehozását, amelyek a fő vonal mellett futnak (*branch*), valamint az ágak összeillesztését (*merge*)



### A szoftverfejlesztés technikai támogatása

#### Verziókövető rendszerek

- az összeillesztés rendszerint utólagos manuális korrekciót igényel
- az összeillesztésnek rendszerint automatikusan illeszti a módosított tartalmakat kódelemzést használva, ez lehet 2 pontos (*two-way*), amikor csak a két módosítást vizsgálja, vagy 3 pontos, amikor az eredeti fájlt is
- programrészek zárolását (*lock*), hogy a konfliktusok kizárhatóak legyenek
- adott verzió, mint pillanatkép (*snapshot*) rögzítése (*tag*), amelyhez a hozzáférés publikus
- feltöltések atomi műveletként történő kezelését (pl. megszakadó feltöltés esetén visszavonás)

## A szoftverfejlesztés technikai támogatása

### Verziókezelés a projektben

- A legtöbb szoftverfejlesztési projekt számtalan fejlesztési ágat vezet be a munka során
  - a fő fejlesztési ág a stabil, ellenőrzött és jóváhagyott funkcionalitást tartalmazza
  - a hosszú távú fejlesztési ágak (*long-running branch*) egy adott fejlesztők, vagy egy adott programverzióval kapcsolatos funkcionalitást tartalmaznak
  - a feladat specifikus ágak (*topic branch*) egy konkrét feladathoz tartozó funkcionalitást tárol, annak lefejlesztését követően már nem él
  - a teszt ágak (*test branch*) az ágak közötti összeillesztés hatásait vizsgálják

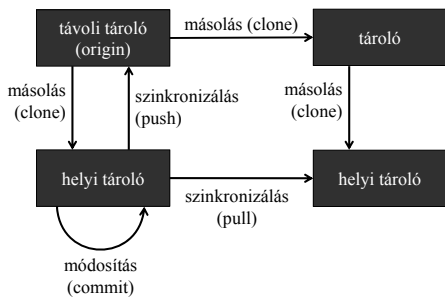
## A szoftverfejlesztés technikai támogatása

### Elosztott verziókövető rendszerek

- A központosított verziókezelők mellett elterjedtek az *elosztott (distributed)* rendszerek
  - pl. *Bazaar, Mercurial, Git*
  - nagyobb szabadságot adnak a tárolók kezelésében, a szinkronizálási lehetőségekben
  - egy főtároló (*origin*) a kiindulási pont, amelyből tetszőleges másolatot készíthetünk (*clone*)
  - a módosítások csak a lokális másolatot befolyásolják, külön kell feltöltenünk azokat valamely távoli tárolóra (*push*), vagy jelölhetjük őket feltöltésre (*pull request*)
  - az egyes tárolók állapotait szinkronizálhatjuk (*pull, fetch*)

## A szoftverfejlesztés technikai támogatása

### Elosztott verziókövető rendszerek



## A szoftverfejlesztés technikai támogatása

### Folyamatos integráció

- A *folytonos integráció (continuous integration, CI)* egy olyan gyakorlati módszer, amely lehetővé teszi a programkódok ellenőrzésének és tesztelésének felgyorsítását
  - célja a lehetséges hibák, integrációs problémák azonnali, automatizált kiszűrése, visszajelzés a fejlesztőnek
  - a programkódok verziókezelő rendszer segítségével egy központi tárhelyre kerülnek, naponta többször
  - a tárhely tartalma minden módosítást követően automatikusan fordításra kerül (*build automation*), a fordítással pedig a lekódolt tesztek is végrehajthatódnak
  - az így ellenőrzött kódot további tesztelés követheti

## A szoftverfejlesztés technikai támogatása

### Folyamatos integráció

