

A satellite is shown in space, with a view of Earth's surface. The satellite has several solar panels and a circular antenna. The background is a light blue and white gradient representing the Earth's atmosphere and surface.

**Eötvös Loránd Tudományegyetem  
Informatikai Kar**

# **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

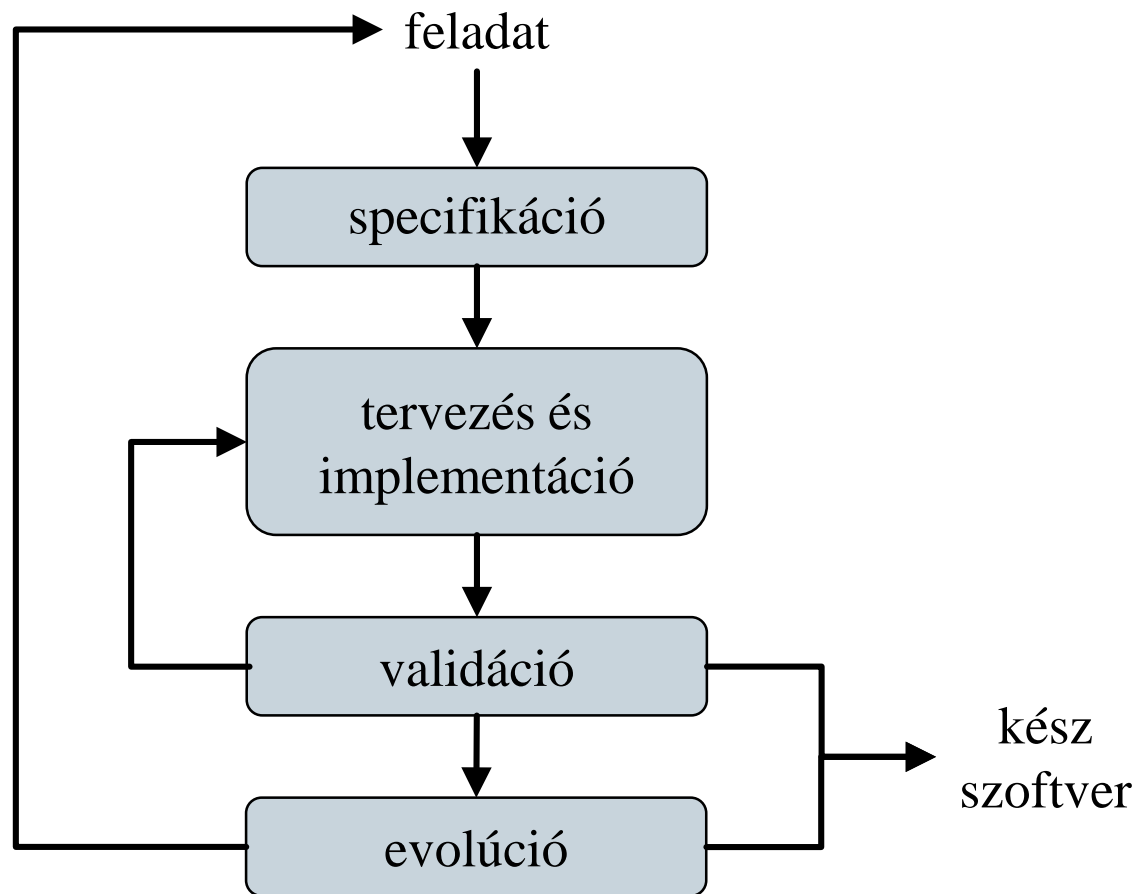
## A szoftver életciklus

---

- 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 előállítása
  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 bárhonnán 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 kinyithatóak
  - a cédulák biztosítják a fejlesztési és tesztelési folyamat naplózását

# A szoftverfejlesztés technikai támogatása

## A Trac projektmenedzser

The screenshot shows the Trac web interface. At the top left is the Trac logo and the text "Integrated SCM & Project Management". Below the logo is a navigation bar with links for Wiki, Timeline, Roadmap, Browse Source, View Tickets, and New Ticket. The main content area is titled "{8} Active Tickets, Mine first (22 matches)". Below this title are two bullet points: "List all active tickets by priority" and "Show all tickets owned by the logged in user in a group first." Below the bullet points is a table of active tickets. The table has columns for Ticket, Summary, Component, Version, Milestone, Severity, Owner, and Created. The first few rows are highlighted in yellow.

Ticket	Summary	Component	Version	Milestone	Severity	Owner	Created
#9	Improved group management interface	webui	0.0.30	advancedu	enhancement	tv*	10/01/04
#4	Speed up webui unit tests	webui	0.0.30	advancedu	normal	tv*	10/01/04
#2	Support STARTTLS as server	misc	0.0.30	server	enhancement	tv*	10/01/04
#21	IdapSyntax API: add e.assume() as a way to edit unknown attributesets	misc			enhancement	tv	01/09/05
#28	URL and I18n of webui	webui			enhancement	tv	02/20/05
#16	Server-side support for service-specific passwords	server			normal	tv	11/21/04
#18	Add support for requiring group membership in ldapator checkers	misc			normal	tv	11/25/04
#19	Add support for a checker that does not pass the resulting entries as avatars	misc			normal	tv	11/25/04
#23	Webui should handle LDAP errors and display them in a prettier way, not as tracebacks	webui			normal	tv	02/04/05
#24	Fail nicer with invalid search-field config entries	misc			normal	tv	02/04/05
#25	webui adding inetOrgPerson raises due to case differences in x500nameidentifier	webui			normal	tv	02/04/05
#3	Speed up ldapator-ldap2maradns	utils	0.0.30	advancedu	minor	tv*	10/01/04
#20	apps/webui/search.py:236 render_form() patch	webui		bugfix	normal	tv	09/29/05
#10	Entry per file LDIF storage support	misc	0.0.30	server	enhancement	tv*	10/01/04
#26	Webui should have a way to change the RDN of entry	webui			normal	tv	02/09/05
#27	Webui add should choose better RDNs	webui			trivial	tv	02/09/05
#7	Add utility ldap-missing-homes, list LDAP accounts whose home dir does not exist	utils	0.0.30	advancedu	enhancement	tv*	10/01/04
#11	Add serverside in-memory database with LDIF journal support	server	0.0.30	server	enhancement	tv*	10/01/04
#8	Generate passwords with passgova instead of pwgen	misc	0.0.30	None	enhancement	tv*	10/01/04
#13	Client-side OTP (One Time Password) support	misc	None	None	enhancement	tv*	10/22/04
#14	Server-side OTP (One Time Password) support	misc	None	None	enhancement	tv*	10/22/04
#30	Server-side OTP (One Time Password) support	misc	None	None	enhancement	tv*	10/22/04

The screenshot shows the Trac ticket details page for TKT\_2. At the top is a navigation bar with links for Dashboard, Tickets, and Admin. Below the navigation bar is the title "List of tickets > Ticket details". The main content area is titled "Tickets: TKT\_2" and has buttons for New, Edit, Copy, and Delete. Below the buttons is a table with columns for TicketId, Summary, Type, Priority, Milestone, Status, Keywords, Assigned to, and Reported by. The table contains one row for TKT\_2. Below the table is a section for Attachments with a button for "Attach a new file" and a list of attachments. Below the attachments is a section for Change history with a button for "Change history" and a list of changes.

TicketId	Summary	Type	Priority	Milestone	Status	Keywords	Assigned to	Reported by
TKT_2	test openbiz trac	Defect	Medium	alpha	Open			

Attachments:

- favicon.ico (3638 bytes) added by test on 2008-02-18 17:06:50  
Description: aaaa
- 106746kg.rar (33047 bytes) added by rocky on 2008-02-18 17:08:30  
Description: replace file test

Change history:

2008-02-18 17:26:36 changed by rocky\_test

The error I'm referring to in comment 26 is an exception raised in a WikiSyntaxProvider, inside a wiki formatting done while rendering the timeline. The corresponding traceback for that error can be seen in attachment timeline\_rendering\_backtrace.log

if I catch the exception before it goes through the template rendering, the memory usage is OK again. The live object count, as measured by len(gc.get\_objects()) after a gc.collect(), remains constant at 62642, after successive requests.

raise an exception on purpose after the template rendering, there's also no increase in memory usage.

If the original exception is not caught, the live object count increase by ~50000 after each call.

try to locate the exact point in the traceback which cause the memory retention.

02-13 01:50:59 changed by rocky

description: changed from 1234567890 to Build trac with openbiz  
asdsad - learn the trac data model (table)

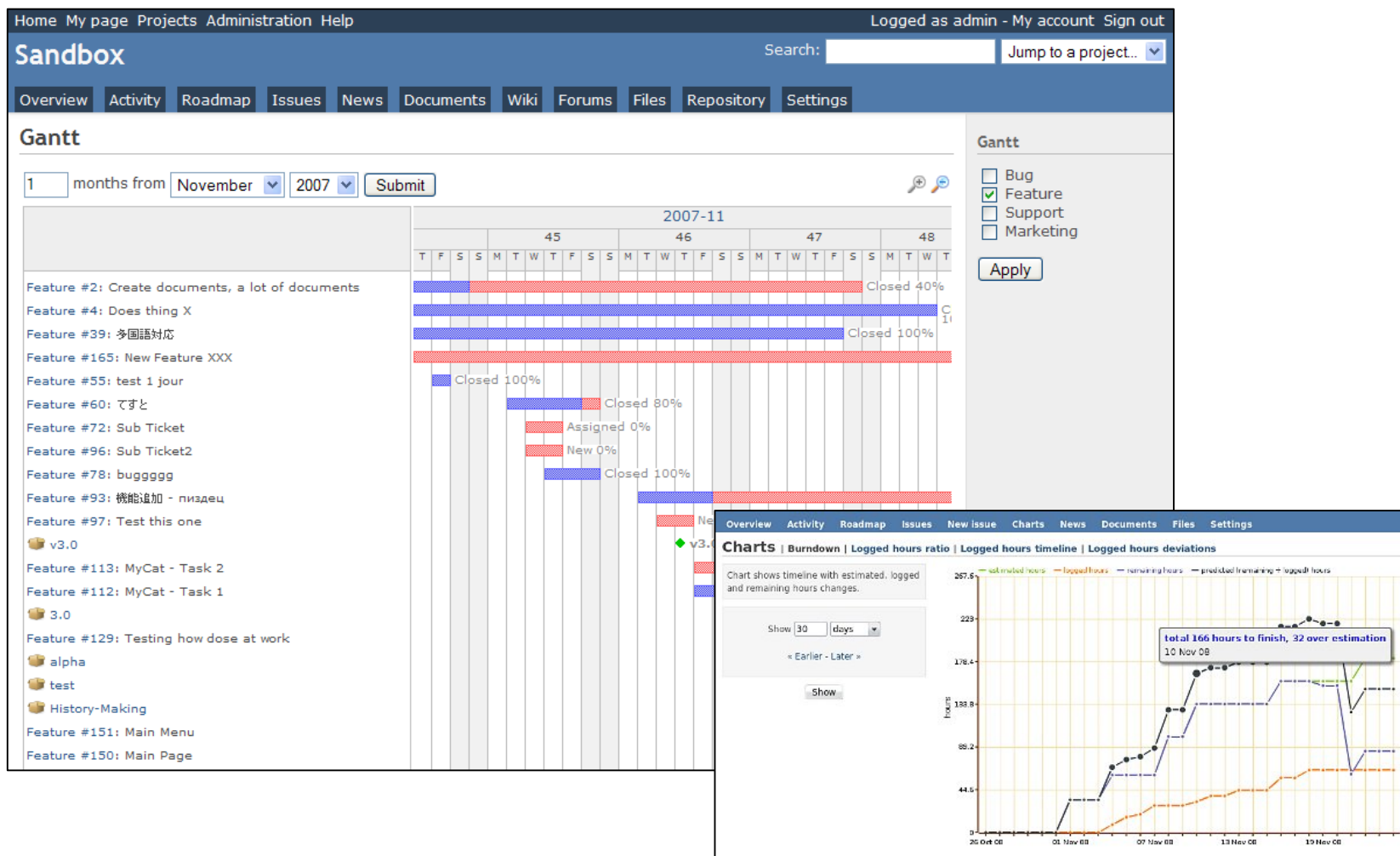
The screenshot shows the Trac file browser interface. At the top left is the Trac logo and the text "Integrated SCM & Project Management". Below the logo is a navigation bar with links for Wiki, Timeline, Roadmap, Browse Source, View Tickets, and New Ticket. The main content area is titled "root / organizer / trunk". Below the title is a table of files and directories. The table has columns for Name, Size, Rev, Age, and Last Change. The first few rows are highlighted in yellow.

Name	Size	Rev	Age	Last Change
../				
icons		6	21 hours	root: Moved remotel
page		22	21 hours	root: fixed wrong fun
changelog.txt	1.5 kB	8	21 hours	root: Moved remotel
class_filereader.php	2.3 kB	9	21 hours	root: Moved remotel



# A szoftverfejlesztés technikai támogatása

## A Redmine projektmenedzser



# A szoftverfejlesztés technikai támogatása

## Projektmenedzsment 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

# A szoftverfejlesztés technikai támogatása

## Projektvezető szolgáltatások

---

- A projektvezető szolgáltatások (*project hosting services*) általában rendelkezésre bocsátanak több projektfejlesztő eszközt
  - projektmenedzsment, 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*)

# 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 projektmenedzsment eszközhöz

# 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 egységteszteket)

# A szoftverfejlesztés technikai támogatása

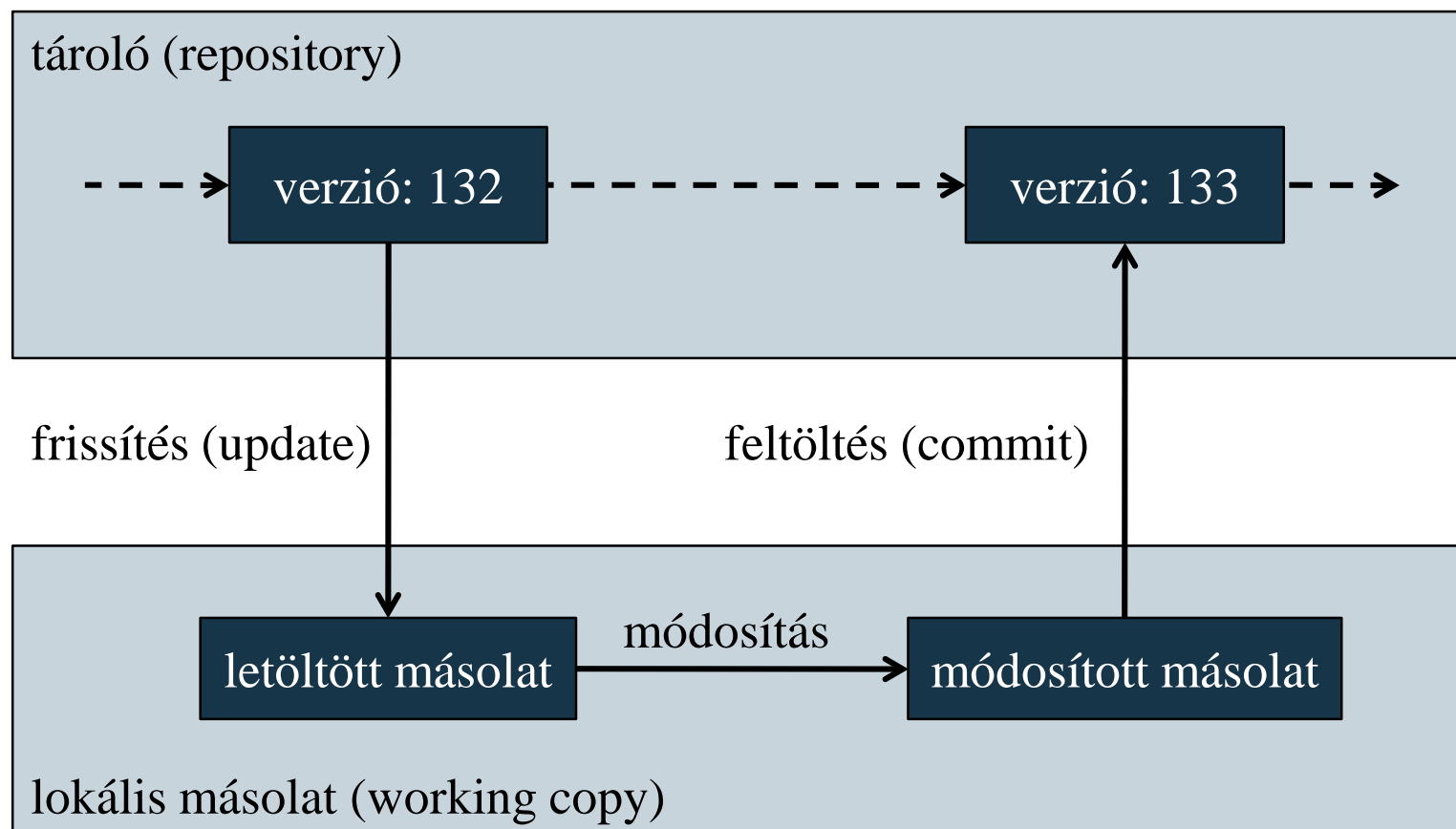
## Verziókövető rendszerek

---

- Mivel az implementáció több lépésben, és sokszor párhuzamosan zajlik, szükséges, hogy az egyes programállapotok, jól követhetők 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 visszatöltik 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

---

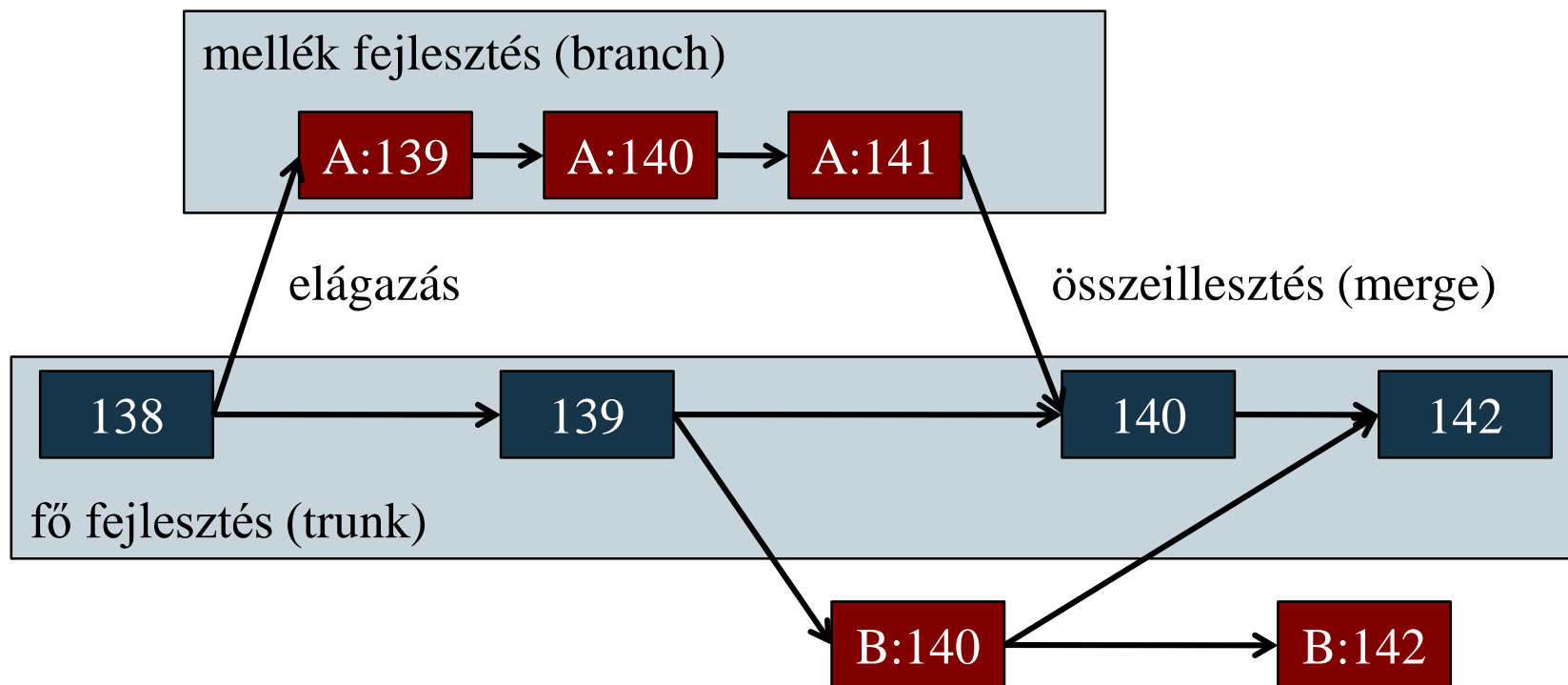
- A rendszerek lehetővé teszik:
  - az összes eddig változat (*revision*) eltárolását, illetve annak letöltési lehetőségét
  - a fő fejlesztési vonal (*baseline, master* vagy *trunk*) és a legfrissebb változat (*head*) elérését, új változat feltöltését annak dokumentálásával
  - az egyes változatok közötti 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

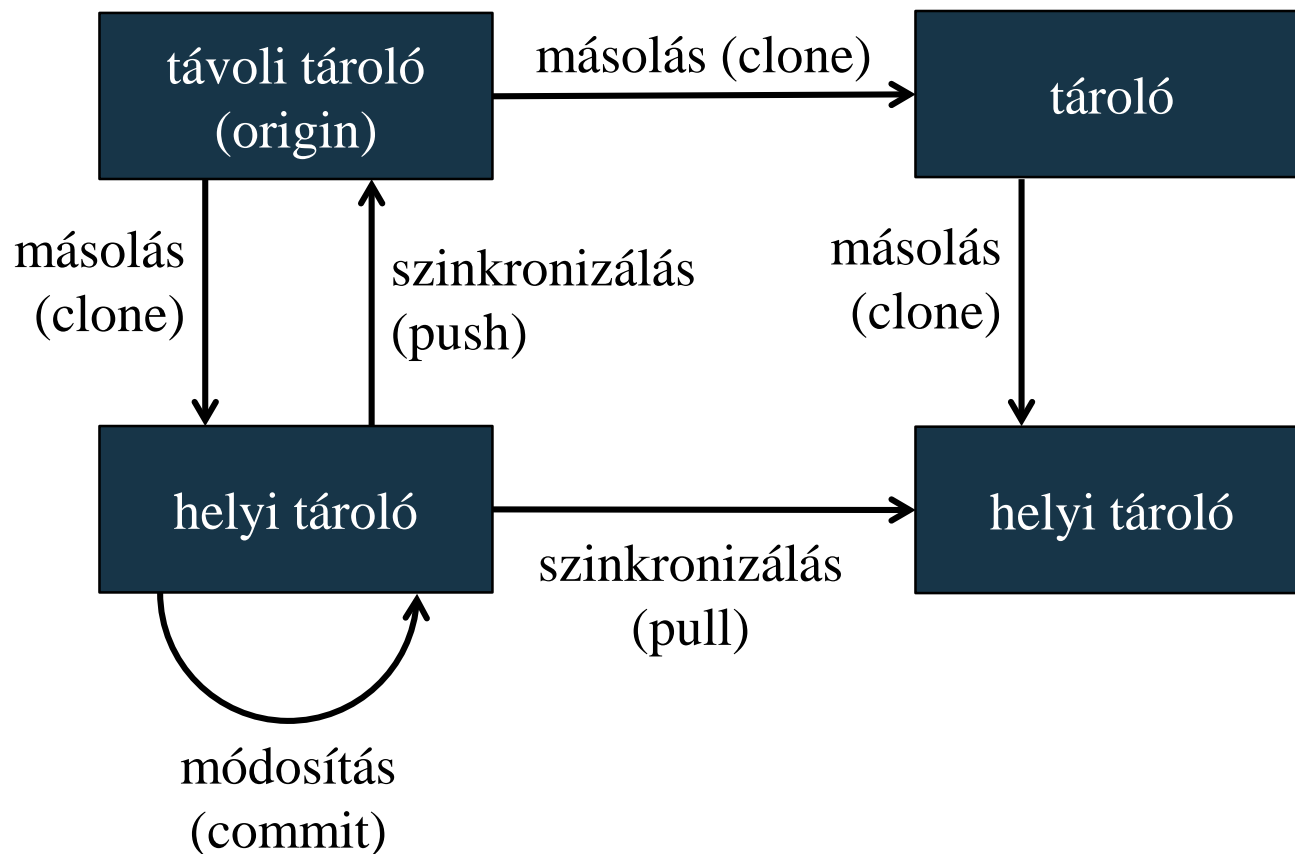
## Elosztott verziókövető rendszerek

---

- 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 végrehajthatók
  - az így ellenőrzött kódot további tesztelés követheti

# A szoftverfejlesztés technikai támogatása

## Folyamatos integráció

