



Programozási technológia 2.

Projektmenedzsment eszközök,
Verziókövető rendszerek

Cserép Máté
ELTE Informatikai Kar
2019.

Projektmenedzsment eszközök

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

Projektmenedzsment eszközök

Funkcionalitás

- ▶ 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

Projektmenedzsment eszközök

Ütemterv és időzítés

- ▶ A szoftver lehetőséget ad, hogy a projekt ütemtervét elkészítsük, és azt folyamatosan szem előtt tarthassuk
 - ▶ definiálhatunk mérföldköveket (*milestone*), 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

Projektmenedzsment eszközök

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

Projektmenedzsment eszközök

Példák

- ▶ 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
 - ▶ *YouTrack*: a JetBrains rendszere, Java alapokon, Xodus adatbázis háttérrel (NoSQL)

Projektmenedzsment eszközök

A Trac projektmenedzser

trac
Integrated SCM & Project Management

Wiki | Timeline | Roadmap | Browse Source | View Tickets | New Ticket | Search

{8} Active Tickets, Mine first (22 matches)

- List all active tickets by priority.
- Show all tickets owned by the logged in user in a group first.

Active Tickets

Ticket	Summary	Component	Version	Milestone	Severity	Owner	Created
#9	Improved group management interface	webui	0.0.30	advancedui	enhancement	tv *	10/01/04
#4	Speed up webui unit tests	webui	0.0.30	advancedui	normal	tv *	10/01/04
#2	Support STARTTLS as server	misc	0.0.30	server	enhancement	tv *	10/01/04
#21	Idap syntax API: add e.assume() as a way to edit unknown attributesets	misc			enhancement	tv	01/09/05
#28	URL and IDN of webui	webui			enhancement	tv	02/20/05
#16	Server-side support for service-specific passwords	server			normal	tv	11/20/04
#18	Add support for requiring group membership in ldapotr 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 s500uniqueidentifier	webui			normal	tv	02/04/05
#3	Speed up ldapotr-ldap2marads	utils	0.0.30	advancedui	minor	tv *	10/01/04
#29	apps/webui/search.py:236 render_form() patch	webui			bugfix	normal	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	advancedui	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 support	misc	None	None	enhancement	tv *	10/22/04
#30	Server-side OTP support	misc	None	None	enhancement	tv *	10/22/04

Dashboard | Tickets | Admin |

List of tickets > Ticket details

Tickets: TKT_2

New | Edit | Copy | Delete

Ticket Id: TKT_2 | Reported by:

Summary: test openbiz trac

Type: Defect | Component: BizForm

Priority: Medium | Severity: Critical

Milestone: alpha | Version: 1.0

Status: Open | Resolution:

Keywords: CC:

Assigned to:

Description: Build trac with openbiz
- learn the trac data model (table)
- build dataobis, forms, views
- write templates and css

Attachments

Attach a new file

- favicon.ico (3638 bytes) added by test on 2008-02-18 17:06:50
Description: *aaaa*
- 106748Kg.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 backtrace for that error can be seen in attachment:timeline_rendering_backtrace.log

Now, 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.

When I 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.

by to locate the exact point in the backtrace which cause the memory retention.

02-13 01:50:59 changed by rocky

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

trac
Integrated SCM & Project Management

logged in as root | Logout | Settings | Help

Wiki | Timeline | Roadmap | Browse Source | View Tickets | New Ticket

root / organizer / trunk

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

Projektmenedzsment eszközök

A Redmine projektmenedzser

The screenshot displays the Redmine Gantt chart interface. At the top, there is a navigation bar with links for Home, My page, Projects, Administration, and Help. The user is logged in as 'admin' and can access My account or Sign out. The main header shows 'Sandbox' and a search bar. Below the header, there are tabs for Overview, Activity, Roadmap, Issues, News, Documents, Wiki, Forums, Files, Repository, and Settings. The Gantt chart is titled 'Gantt' and shows a timeline for 2007-11. The chart is divided into four weeks, with days of the week (T, F, S, S, M, T, W, T, F, S, S, M, T, W, T) listed above. The tasks are listed on the left, and their progress is shown as horizontal bars. The tasks and their progress are:

- Feature #2: Create documents, a lot of documents (Closed 40%)
- Feature #4: Does thing X (Closed 100%)
- Feature #39: 多言語対応 (Closed 100%)
- Feature #165: New Feature XXX (Closed 100%)
- Feature #55: test 1 jour (Closed 100%)
- Feature #60: 飞すと (Closed 80%)
- Feature #72: Sub Ticket (Assigned 0%)
- Feature #96: Sub Ticket2 (New 0%)
- Feature #78: buggggg (Closed 100%)
- Feature #93: 機能追加 - пиздец (Closed 100%)
- Feature #97: Test this one (New 0%)
- v3.0 (Closed 100%)
- Feature #113: MyCat - Task 2 (Closed 100%)
- Feature #112: MyCat - Task 1 (Closed 100%)
- 3.0 (Closed 100%)
- Feature #129: Testing how dose at work (Closed 100%)
- alpha (Closed 100%)
- test (Closed 100%)
- History-Making (Closed 100%)
- Feature #151: Main Menu (Closed 100%)
- Feature #150: Main Page (Closed 100%)

On the right side, there is a legend for the Gantt chart with the following items:

- Bug
- Feature
- Support
- Marketing

An 'Apply' button is located below the legend. The chart also shows a 'Submit' button and a '1 months from' dropdown menu set to 'November' and '2007'.



Projektmenedzsment eszközök

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.: *GitHub*, *GitLab*, *SourceForge*, *Bitbucket*
 - ▶ egyes szolgáltatások bizonyos programozási nyelvek, vagy témakör köré csoportosulnak (pl. *mozdev*)

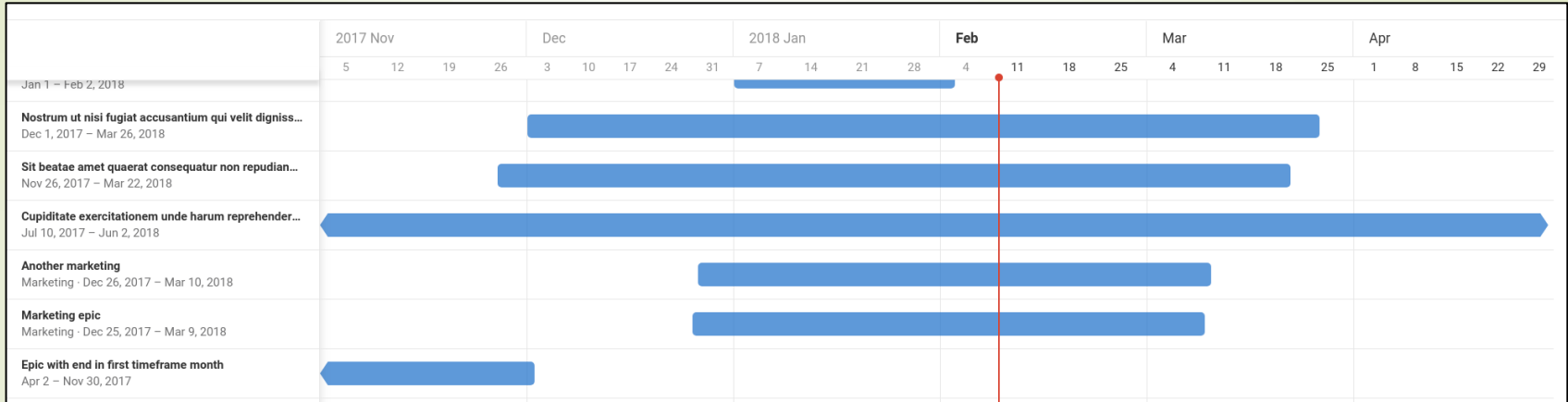
Projektmenedzsment eszközök

A GitLab projektvezető szolgáltatás

- ▶ A GitLab egy webes felülettel rendelkező, számos szoftvereszközt integráló projektvezető szolgáltatás
 - ▶ Használható a gitlab.com oldalon elérhető keresztül (SaaS)
 - ▶ hasonló a GitHub-hoz
 - ▶ ingyenes, korlátlan (privát) projekttel (előfizetés elérhető)
 - ▶ Gitlab Community Edition
 - ▶ saját szerveren kiszolgált (*self-hosted*)
 - ▶ megfelelő kisebb és közepes méretű csapatok számára
 - ▶ ingyenes, nyílt forráskódú
 - ▶ GitLab Enterprise Edition
 - ▶ extra funkciókkal, amelyekre jellemzően nagyobb fejlesztői állomány (100+ fő) esetén van szükség

Projektmenedzsment eszközök

GitLab - Projektmenedzsment



GitLab / GitLab.org

Milestones

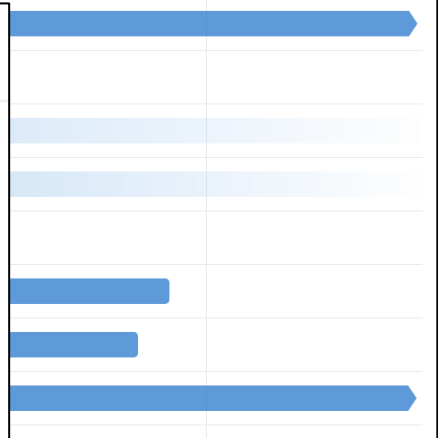
Open 24 Closed 86 All 102

10.1 - Group Milestone 0% complete
8 Issues · 0 Merge Requests

10.2 - Group Milestone 0% complete
0 Issues · 0 Merge Requests

8.17 - Project Milestone 100% complete
2 Issues · 4 Merge Requests
`gitlab-shell` `gitlab-workhorse` `gitlab-pages`

9.1 - Project Milestone 100% complete
3 Issues · 6 Merge Requests
`gitlab-elasticsearch-indexer` `gitlab-pages` `gitlab-shell`
`gitlab-workhorse`



Projektmenedzsment eszközök

GitLab - Feladatkövetés

The screenshot displays the GitLab Issues interface. At the top, it shows the navigation bar with 'GitLab / GitLab.org / GitLab Community Edition' and buttons for 'Edit Issues' and 'New Issue'. Below this, there are statistics for 'Open' (8,469), 'Closed' (17,917), and 'All' (26,386) issues. A search bar is present with the text 'Search or filter results...' and a 'Last updated' dropdown menu.

The main content area shows a list of issues. The first issue is 'Emoji picker is too aggressive' (#35925) by Bence Nagy, updated 4 minutes ago. Below it, there are several task cards, each with a title, a description, and various labels. The tasks are organized into columns, each with a title and a count of items:

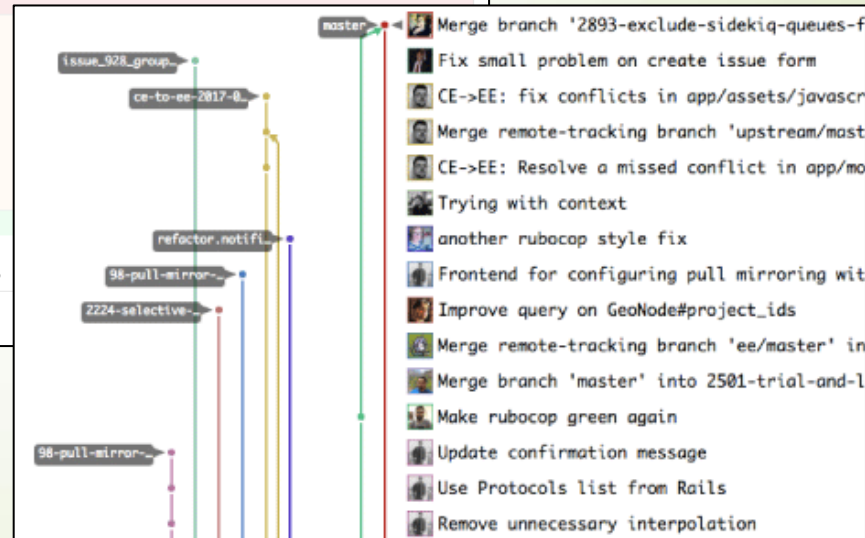
- Backlog** (5 items):
 - Remove ghost notification settings for groups and projects (gitlab-org/gitlab-ce#44824) - 3 items. Labels: Plan, Stretch, backend, database, notifications, technical debt.
 - Create RuboCop cop for url_for(params, ...) (gitlab-org/gitlab-ce#47986) - 1 item. Labels: Plan, backend, backstage, security, static analysis.
 - Consider updating all API endpoints that accept comma-separated strings to also accept arrays (gitlab-org/gitlab-ce#48007) - 1 item. Labels: Plan, Stretch, api, backend, enhancement.
 - Use serializer to render diff lines (gitlab-org/gitlab-ce#48084) - 2 items. Labels: Plan, backend, diff, technical debt.
 - Stored XSS on issue details page (gitlab-org/gitlab-ce#49422) - 1 item. Labels: HackerOne, P2, Plan, S2, frontend, issues, markdown, security.
- performance** (5 items):
 - Controller Projects::MergeRequestsController#ci_environments_status executes more than 100 SQL queries (gitlab-org/gitlab-ce#43109) - 1 item. Labels: P3, Plan, S3, backend, database, performance.
 - SQL timings in Projects::MergeRequestsController#show.json should be no more than 100ms per request (gitlab-org/gitlab-ce#43394) - 1 item. Labels: P3, Plan, S2, backend, database, performance.
 - Cache diff related data from blobs when loading "/diffs" (gitlab-org/gitlab-ce#45693) - 1 item. Labels: Gitaly, Plan, Stretch, backend, diff, performance.
 - SQL timings for Projects::IssuesController#show should be below 100 ms per request (gitlab-org/gitlab-ce#48220) - 1 item. Labels: Plan, S2, backend, database, issues, performance.
 - Limit the number of comments on a notable (gitlab-org/gitlab-ce#46676) - 1 item. Labels: Deliverable, Plan, backend, issues, merge requests, performance.
- bug** (7 items):
 - Postgres timeout when counting number of CI builds for usage ping (gitlab-org/gitlab-ce#45938) - 1 item. Labels: Plan, admin dashboard, backend, bug, reproduced on GitLab.com, usage ping.
 - Searching in a group does not search subgroups (gitlab-org/gitlab-ce#47395) - 1 item. Labels: Plan, backend, bug, search, subgroups.
 - Can't attach image to epic description and comment (gitlab-org/gitlab-ee#7009) - 1 item. Labels: Deliverable, Plan, backend, bug, due-22nd, epics, portfolio management.
 - Stack trace of error from uploading image for object storage spews into screen (gitlab-org/gitlab-ee#6699) - 1 item. Labels: Object Storage, Plan, backend, bug.
 - Better error handling for Elastic (gitlab-org/gitlab-elasticsearch-indexer#19) - 1 item. Labels: Deliverable, Plan, backend, bug, elasticsearch.
 - All GitLab branches listed in Jira task in Development section (gitlab-org/gitlab-ee#6752) - 1 item. Labels: Deliverable, Plan, backend, bug, customer, jira.
 - GitLab does not add comment/link to Jira on mention in commit message - 1 item.
- Deliverable** (22 items):
 - Account for issues created in the middle of a milestone in burndown chart (gitlab-org/gitlab-ee#6174) - 3 items. Labels: Deliverable, Plan, backend, direction, due-22nd, milestones.
 - Order issues / merge requests / epics lists in both directions (gitlab-org/gitlab-ee#39849) - 1 item. Labels: Accepting Merge Requests, Deliverable, Plan, UX ready, backend, due-22nd, frontend, issues, merge requests.
 - Merge request list row design (gitlab-org/gitlab-ce#47010) - 1 item. Labels: Deliverable, Plan, UX ready, direction, due-22nd, frontend, merge requests.
 - Sort by start date and end date in the roadmap view and epics list view (gitlab-org/gitlab-ee#6494) - 1 item. Labels: Deliverable, Plan, UX, backend, due-22nd, epics, frontend, portfolio management, roadmaps.
 - Quick action to add/remove issue to epic from issue (gitlab-org/gitlab-ee#6959) - 1 item. Labels: Deliverable, Plan, UX ready, backend, direction, due-22nd, epics, frontend, issues, portfolio management, quick actions.

Projektmenedzsment eszközök

GitLab – Forráskód menedzsment

The screenshot displays two pull requests in a GitLab interface. The top pull request is for a file named `db/migrate/20130110172407_add_public_to_project.rb`. It shows a diff where a class `RenameOwnerToCreatorForProject` is updated from version 2 to 4.2. A review comment from Victor Wu (@victorwu) is pending, asking for "Whitespace consistency. Please fix." Below the comment are buttons for "Submit review" (with a count of 2) and "Add comment now".

The bottom pull request is for the same file. It shows a diff where a class `AddPublicToProject` is added in version 4.2. A review comment from Victor Wu (@victorwu) is pending, asking "Can you explain why we need to change this part?". Below the comment are buttons for "Submit review" (with a count of 2) and "Add comment now".



Projektmenedzsment eszközök

GitLab – Folytonos integráció

GitLab / GitLab.org / GitLab Community Edition

Pipelines

All 57691 Pending 0 Running 6 Finished 55395 Branches Tags Run Pipeline CI Lint

Status	Pipeline	Commit	Stages	
running	#10572747 by latest	master -> 0ef8fd0d Merge branch 'rs-minor-banzai-perf' int...		
running	#10572744 by latest	35729-user-dro... -> a16a6540 Fixes problem due to bad conversion of...		
running	#10572623 by latest	gitaly-404-com... -> 92df0cfb stupid ruby		
failed	#10571382 by latest	bvl-show-proje... -> 471a64c1 Include star_count in project json for da...		00:50:19 4 minutes ago
failed	#10570982 by latest	30343-projects... -> f095211a Fix project item styling		00:56:26 15 minutes ago
failed	#10570829 by latest	32844-issuabl... -> 36b071e0 Move some after_create parts to worke...		01:06:07 14 minutes ago
failed	#10570408 by latest	28283-uuid-sto... -> fc7c4dca New storage is now "Hashed" instead o...		01:01:28 34 minutes ago
				00:59:18 55 minutes ago

Build Test Staging Production

```
graph LR; subgraph Build; B[build]; end; subgraph Test; T1[test1]; T2[test2]; end; subgraph Staging; S[auto-deploy-ma...]; end; subgraph Production; P[deploy to produ...]; end; B --> T1; B --> T2; T1 --> S; T2 --> S; S --> P;
```

Verziókövető rendszerek

Történeti háttér

- ▶ A szoftverek méretének és komplexitásának növekedésével létrejött szoftverkrízis következményeként megnövekedett:
 - ▶ a programok forráskódjának mérete,
 - ▶ a szoftverprojektek megvalósításához szükséges idő,
 - ▶ és szükséges programozói erőforrás.
- ▶ A szoftveripar fejlődésével egyre több alkalmazás készült
 - ▶ a fejlesztések életciklusa gyakran nem ért véget a program első publikus verziójának kiadásával,
 - ▶ karbantartási és további fejlesztési fázisok követték.
- ▶ **A szoftverprojektek méretben, komplexitásban, időben és a résztvevő fejlesztők számában is növekedni kezdtek.**

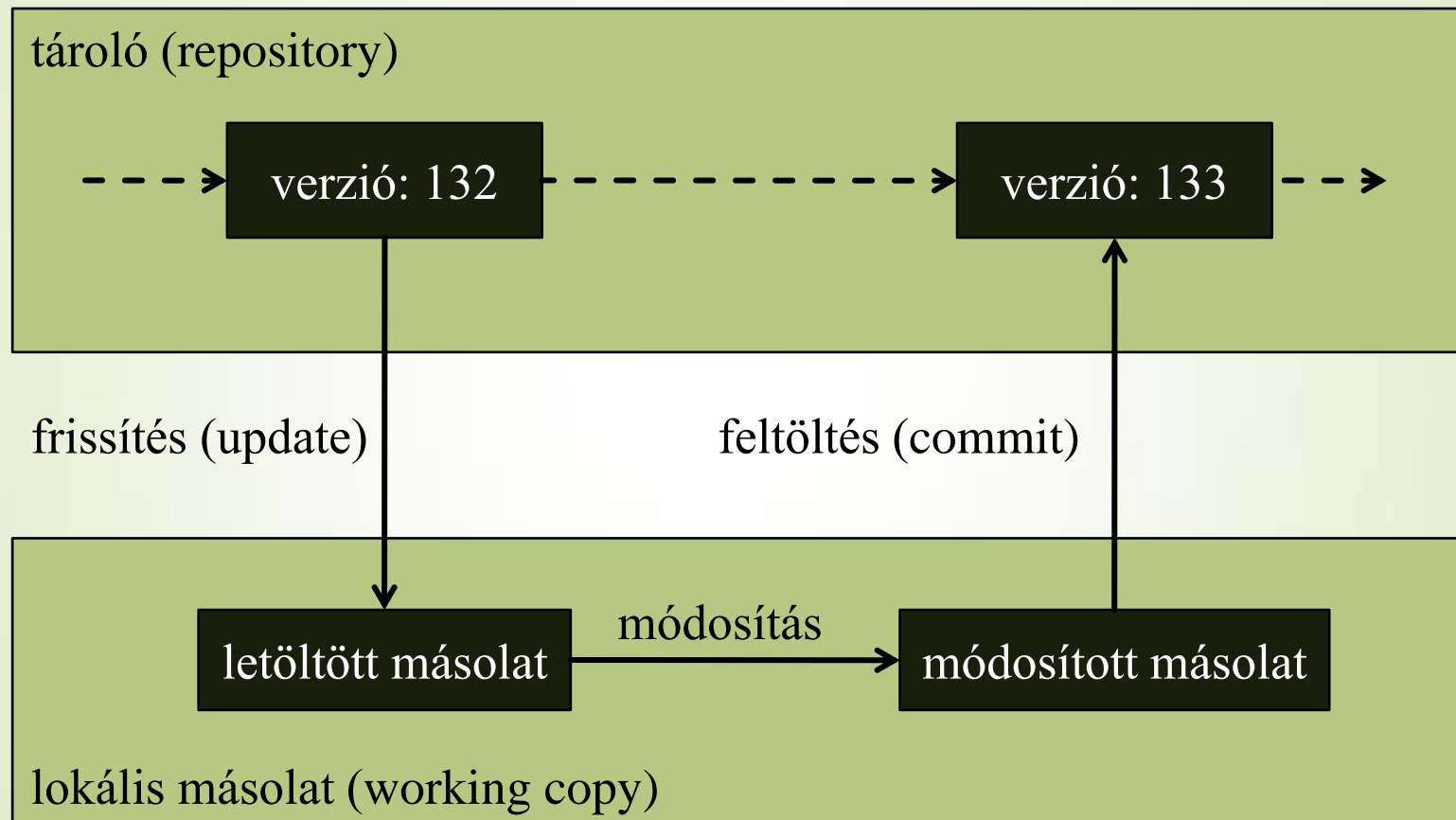
Verziókövető rendszerek

Funkcionalitás

- Mivel az implementáció tehát 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, Git*
 - 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*)

Verziókövető rendszerek

Funkcionalitás



Verziókövető rendszerek

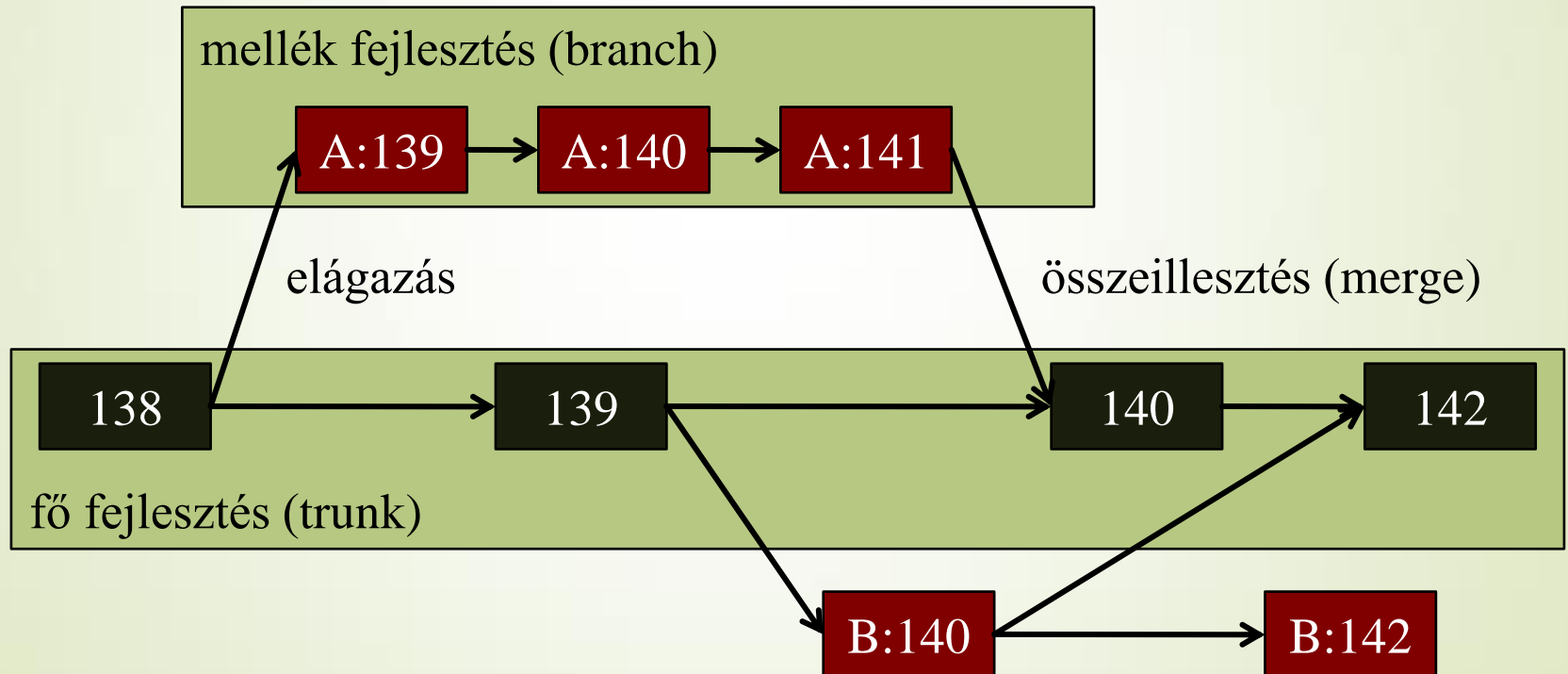
Funkcionalitás

- ▶ A verziókövető 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*)

Verziókövető rendszerek

Funkcionalitás

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



Verziókövető rendszerek

Funkcionalitás

- ▶ 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)

Verziókövető rendszerek

Lokális verziókövető rendszerek (1. generáció)

- ▶ Forráskód változásainak követése, a szoftver funkcióinak különböző kombinációjával készült kiadások kezelése
 - ▶ lokális tároló (de többen is elérhetik pl. *mainframe* esetén)
 - ▶ fájl alapú műveletvégzés (1 verzió 1 fájl változásai)
 - ▶ konkurenciakezelés kizárólagos zárok által
- ▶ Az 1970-es években lefektetésre kerültek az elméleti alapok
 - ▶ Source Code Control System (SCCS) – 1972
 - ▶ Revision Control System (RCS) - 1982

Verziókövető rendszerek

Centralizált verziókövető rendszerek (2. generáció)

- ▶ Több fejlesztő általi párhuzamos szoftverfejlesztés támogatásának előtérbe kerülésre
 - ▶ centralizált modellt megtartva, de kliens-szerver architektúra
 - ▶ fájlhalmaz alapú műveletek (1 verzió több fájl változásai)
 - ▶ konkurenciakezelés jellemzően beküldés előtti egyesítéssel (*merge before commit*)
- ▶ Az 1990-es évektől terjedtek el:
 - ▶ Concurrent Versions System (CVS)
 - ▶ Subversion (SVN)
 - ▶ SourceSafe, Perforce, Team Foundation Server, stb.
- ▶ *Hátrány: a szerver kitüntetett szerepe (pl. meghibásodás), továbbá a verziókezeléshez hálózati kapcsolat szükségeltetik*

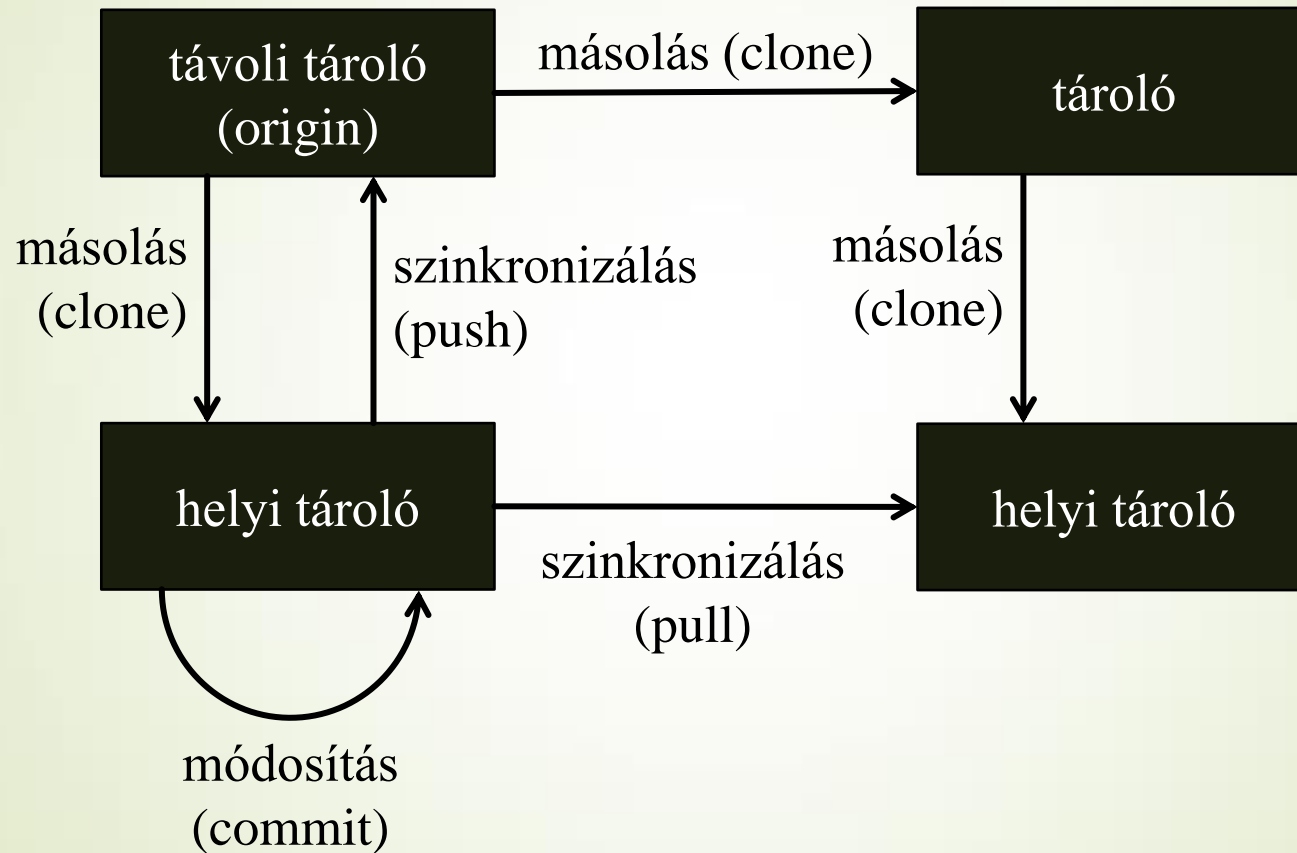
Verziókövető rendszerek

Elosztott verziókövető rendszerek (3. generáció)

- ▶ A klasszikus verziókezelő műveletekről leválasztásra kerül a hálózati kommunikáció, azok a felhasználó által kezdeményezhető önálló tevékenységekként jelennek meg
 - ▶ decentralizált, elosztott hálózati modell
 - ▶ minden kliens rendelkezik a teljes tárolóval és verziótörténettel
 - ▶ a revíziókezelő eszköz műveletei lokálisan, a kliens tárolóján történnek
 - ▶ a kommunikáció *peer-to-peer* elven történik, de kitüntetett (mindenki által ismert) szerverek felállítására van lehetőség
 - ▶ konkurenciakezelés jellemzően beküldés utáni egyesítéssel (*commit before merge*)
- ▶ A 2000-es évek első felében jelent meg:
 - ▶ Monotone, Darcs, Git, Mercurial, Bazaar, stb.

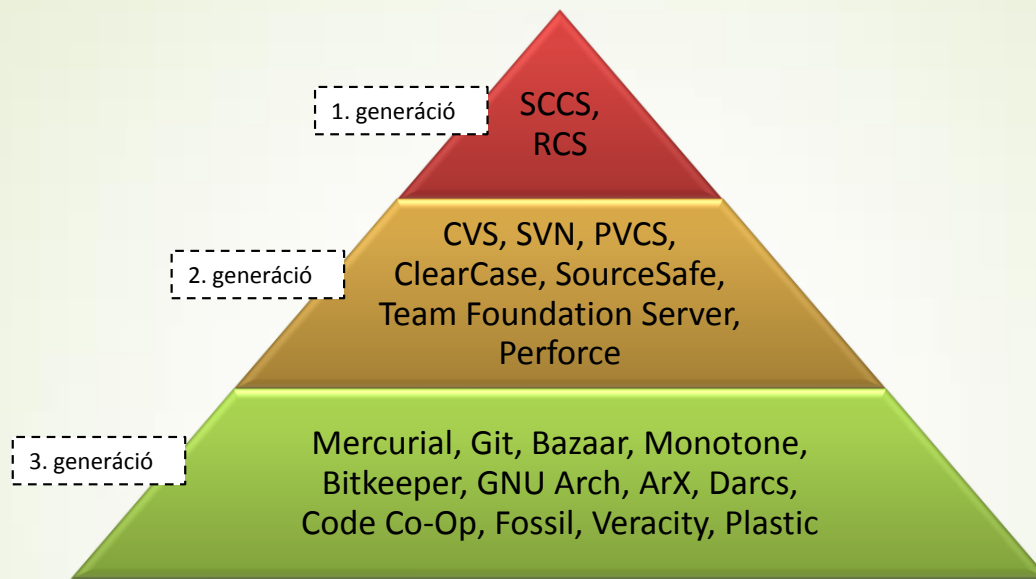
Verziókövető rendszerek

Elosztott verziókövető rendszerek (3. generáció)



Verziókövető rendszerek

Generációs modell



Generáció	Hálózati modell	Műveletvégzés	Konkurenciakezelés
Első	Lokális	Fájlként (non-atomic commits)	Kizórólóagos záruk (exclusive locks)
Második	Központosított	Fájlhalmaz (atomic commits)	Egyesítés beküldés előtt (merge before commit)
Harmadik	Elosztott	Fájlhalmaz (atomic commits)	Beküldés egyesítés előtt (commit before merge)

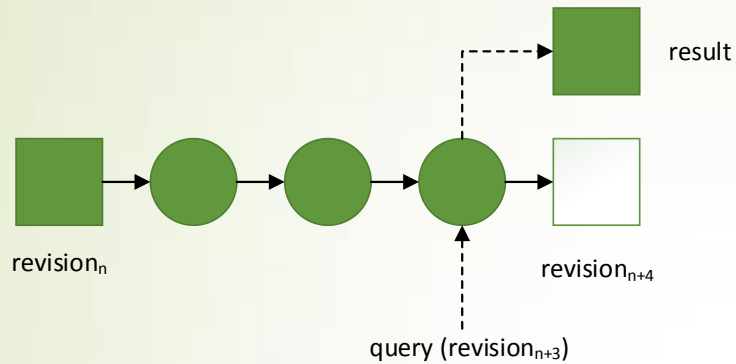
Verziókövető rendszerek

Változások reprezentációja

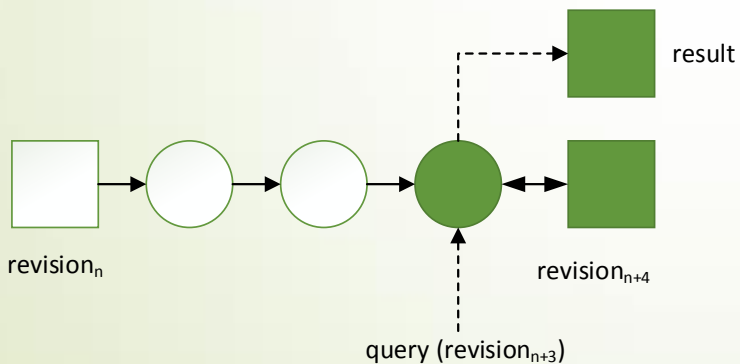
- ▶ A teljes revíziók tárolása nem lehetséges az adattárolás és adatkezelés jelentős költségei miatt
- ▶ A verziókezelő eszközök ezért csak két egymást követő verzió közötti különbséget, a *változáslistát* (*changeset*, *delta*) tárolják
 - ▶ egyes rendszerek (pl. Mercurial) időnként pillanatfelvételt (*snapshot*) készítenek a teljes tartalomról
- ▶ Eleinte (SCCS) a delták a régi verzióból az újat tudták előállítani (*forward deltas*)
- ▶ Korán felmerült (RCS), hogy a fordított delták (*reverse deltas*) használata a legújabb verzió pillanatképének tárolásával jobb teljesítményt nyújthat, ugyanis leggyakrabban egy ág legfrissebb állapotát szokták lekérni
 - ▶ Kevert megoldás is lehetséges, pl. a fő ágon fordított irányú deltákat, a mellékágakon viszont előre mutató delták

Verziókövető rendszerek

Változások reprezentációja



Forward deltas

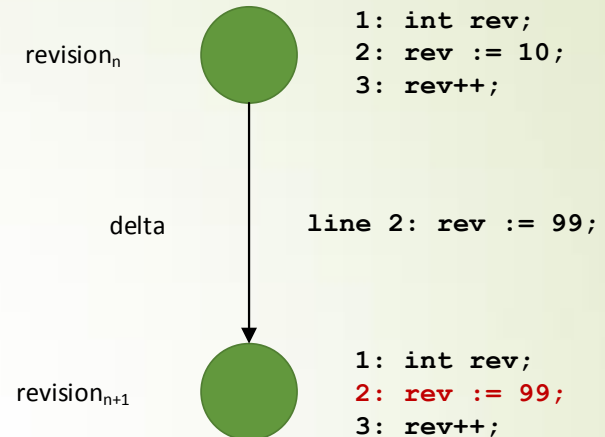


Reverse deltas

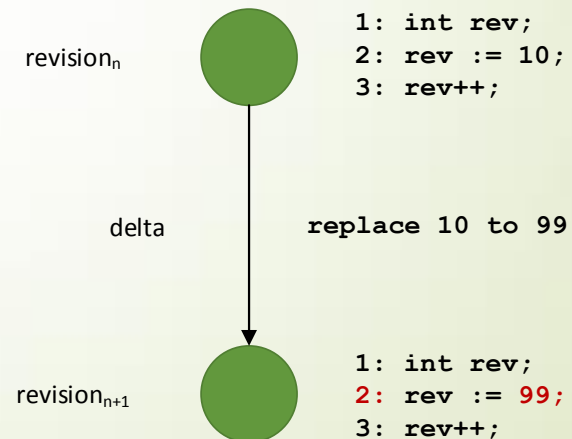
Verziókövető rendszerek

Változások reprezentációja

- ▶ Az eltérések meghatározása szöveges fájlok, így programnyelvi forráskódok esetében jellemzően állapot alapján történik
 - ▶ a legtöbbször soronkénti összehasonlítással
 - ▶ Pl. GNU diff
 - ▶ struktúrált tartalom esetén az összehasonlítás egysége más is lehet (pl. XML, JSON, UML)
- ▶ Bináris adatok (pl. képek) esetén a *művelet alapú* megközelítés is alkalmazható.



Állapot alapú



Művelet alapú

Verziókövető rendszerek

SVN: tároló másolása

```
svn checkout https://mysite.com/best-project
```

```
> A      best-project\branches
```

```
> A      best-project\tags
```

```
> A      best-project\trunk
```

```
> Checked out revision 1.
```

Verziókövető rendszerek

SVN: módosítások beküldése és szinkronizálása

```
cd best-project/trunk
# main.cpp létrehozása
svn add main.cpp
svn commit -m "Added main program."
> Adding          main.cpp
> Transmitting file data .done
> Committing transaction...
> Committed revision 2.
```

Verziókövető rendszerek

SVN: módosítások beküldése és szinkronizálása

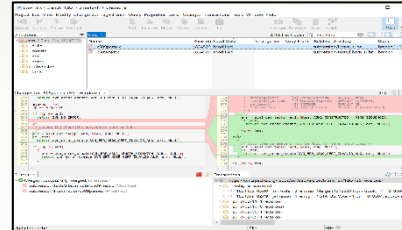
```
# main.cpp módosítása  
svn status  
> M      main.cpp  
svn commit -m "Added main program."  
  
# más kliensek frissítése:  
svn update
```

Verziókövető rendszerek

SVN GUI kliensek

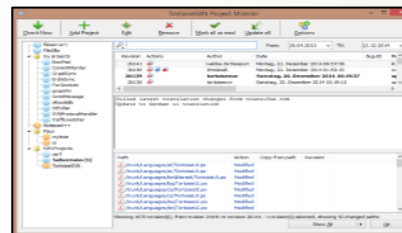
➤ TortoiseSVN

➤ Windows



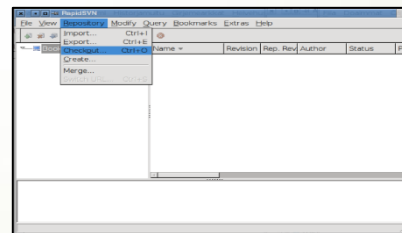
➤ SmartSVN

➤ Linux, Windows, Mac



➤ RapidSVN

➤ Linux, Windows, Mac



Verziókövető rendszerek

Git: tároló másolása

```
git config --global user.name "Hallgató Harold"
```

```
git config --global user.email hallgato@inf.elte.hu
```

```
git clone https://mysite.com/best-project.git
```

```
> Cloning into 'best-project'...
```

```
> remote: Enumerating objects: 3, done.
```

```
> remote: Counting objects: 100% (3/3), done.
```

```
> remote: Total 3 (delta 0), reused 0 (delta 0)
```

```
> Unpacking objects: 100% (3/3), done.
```

Verziókövető rendszerek

Git: módosítások helyi tárolóba küldése

```
cd best-project
# main.cpp létrehozása
git add main.cpp

git status
> On branch master
> Your branch is up to date with 'origin/master'.
> Changes to be committed:
>   (use "git reset HEAD <file>..." to unstage)
>       new file:   main.cpp

git commit -m "Added main program."
> [master d26c7a9] Added main program.
> 1 file changed, 1 insertion(+)
> create mode 100644 main.cpp
```

Verziókövető rendszerek

Git: távoli tárolóval szinkronizálás

```
git push origin master #or simply 'git push'
> Counting objects: 3, done.
> Writing objects: 100% (3/3), 247 bytes | 123.00 KiB/s, done.
> Total 3 (delta 0), reused 0 (delta 0)
> To /path/to/workspace/folder
    d45172c..80a39a2  master -> master
```

```
# main.cpp módosítása
```

```
git add main.cpp
```

```
git commit -m "Edited the main program."
```

```
git push
```

```
# más kliensek frissítése:
```

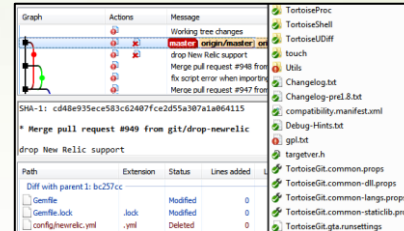
```
git pull
```

Verziókövető rendszerek

Git GUI kliensek

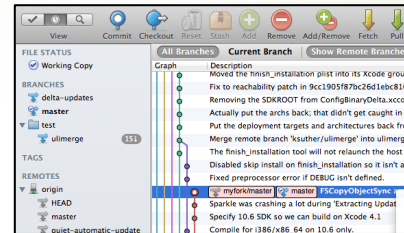
➤ TortoiseGit

➤ Windows



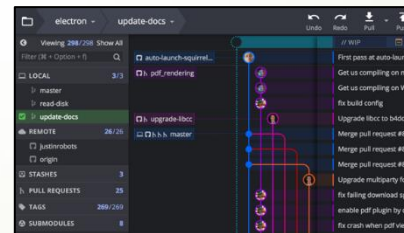
➤ SourceTree

➤ Windows, Mac



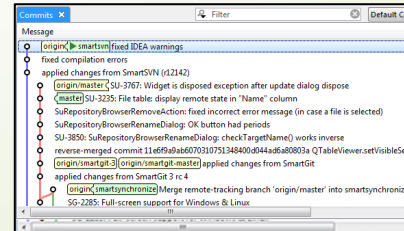
➤ GitKraken

➤ Linux, Windows, Mac



➤ SmartGit

➤ Linux, Windows, Mac



Verziókövető rendszerek

Git Feature Branching Model

- Fő fejlesztési ágak:
 - master
 - develop
- Támogató ágak:
 - feature branches
 - release branches
 - hotfix branches

