

BME MTT 2009

Scherer Balázs, Csordás Péter

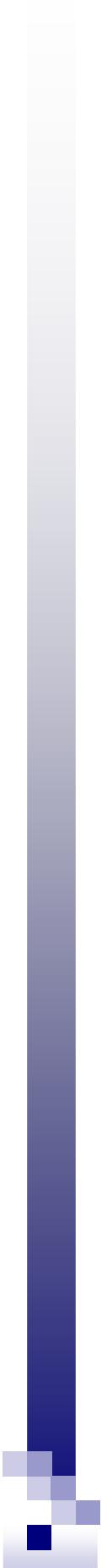
Verziókövetés és a Subversion használata

Bevezetés: Fejlesztés közben fellépő általános problémák

- Egy tipikus fejlesztési probléma:
 - Az eddig futó alkalmazáshoz új kód részletet rakunk
 - Az alkalmazás lefagy
 - Visszaállítjuk a módosításokat
 - Az alkalmazás még minden fagy...
- A helyzet csak bonyolódik, ha nem egyedül dolgozunk:
 - Változtatunk a működő alkalmazáson
 - De valaki más is beleír egy picit
 - Az alkalmazás lefagy

Megoldás: Verziókontroll, verziómenedzsment

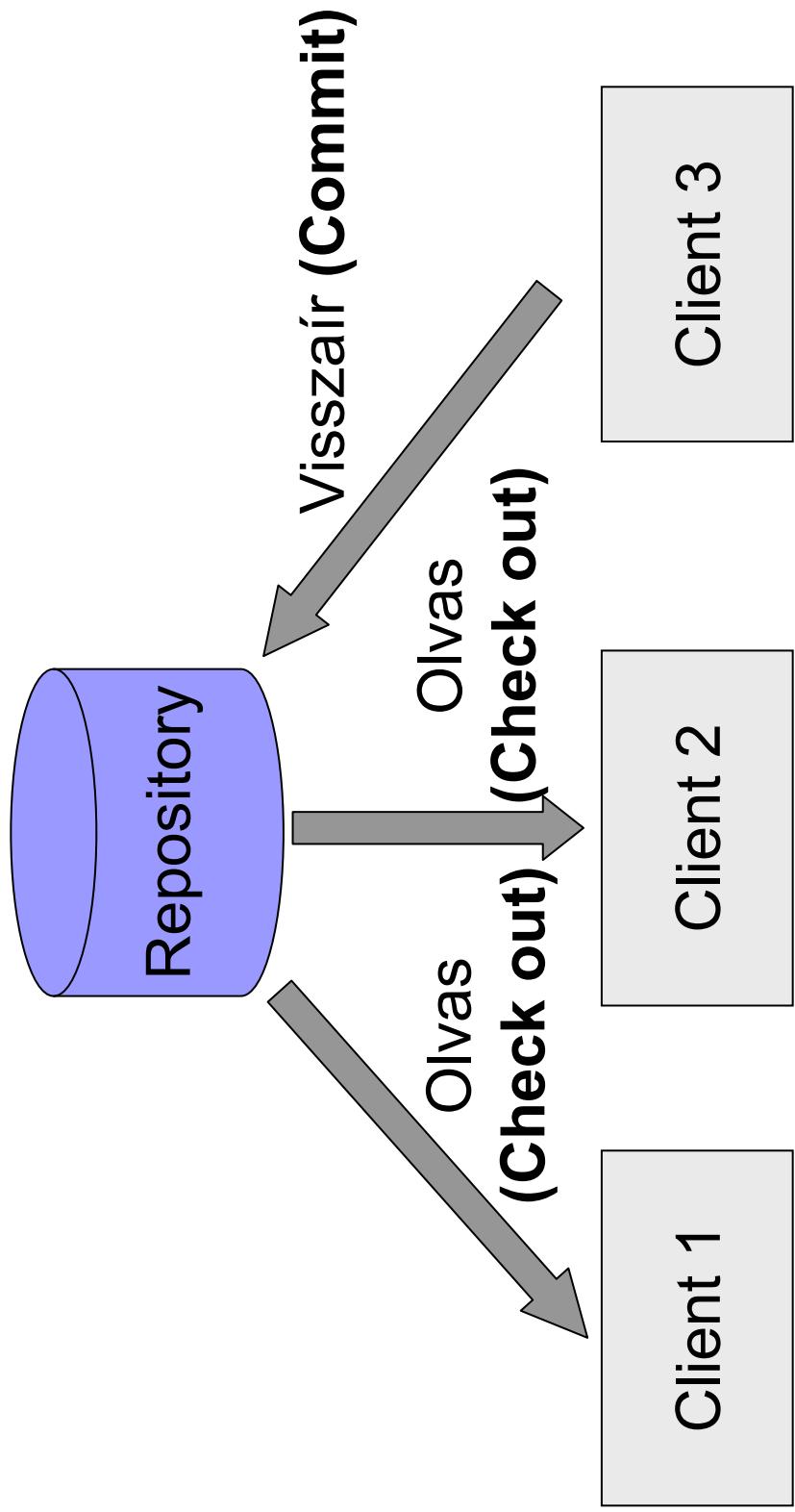
- A verziómenedzsment alapja nem más, mint egy adott project összes változásának nyilvántartása.
- Egy verziómenedzsment rendszer nyilvántart
 - minden egyes file-on létrehozott változást,
 - a könyvtárstruktúrát érintő minden változást.
- A felhasználónak lehetősége van
 - megtekinteni a project vagy egy file állapotát egy adott pillanatban,
 - megtudni, hogy ki, mit és mikor változtatott az adott projecten
 - kommentet tenni minden változtatás mellé



Alapfogalmak: *Repository*, *Client*, *Working copy* (*munkamásolat*)

- Repository (raktár): *Központi nyilvántartása az adatoknak vagy projectnek (a master copy).*
- Client: *Felhasználó, aki dolgozni kíván a projecten.*
- Working copy: *Egy Client által a projectből létrehozott munkaváltozat, amit szabadon változtathat.*

Alapfogalmak: működés



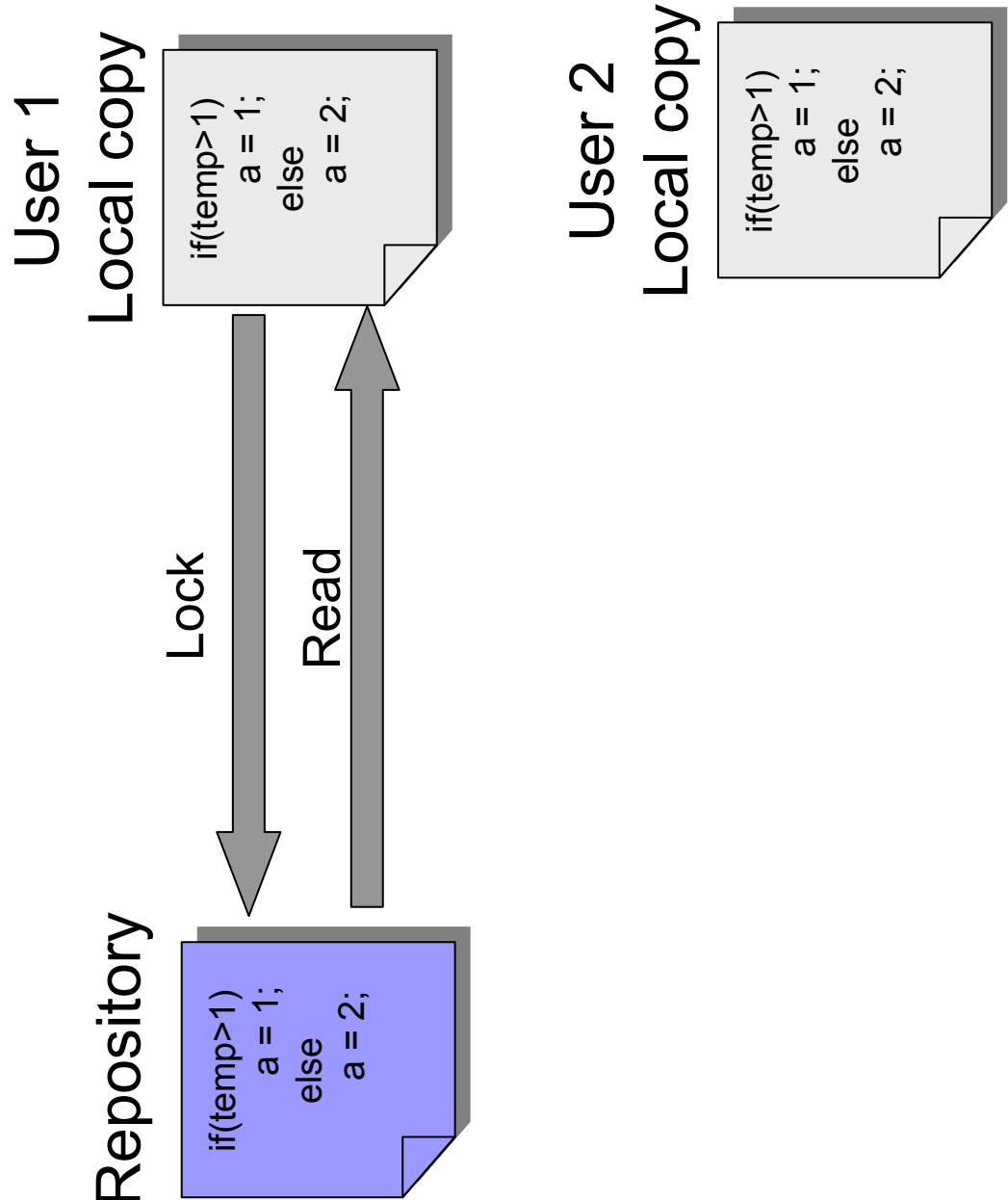
Verziómenedzsment stratégiák: A probléma

- Hogyan támogatja a verziókövető rendszer, hogy a felhasználók együtt dolgozzanak, de mégse lépjenek egymás lábára?
- Ilyen stratégia nélkül könnyen előfordulhat, hogy egy file-t vagy projectet egyszerre többen módosítanak, majd felülírják egymás módosításait (a módosítások nem tünnek el, de nem is kerülnek bele az új verzióba).

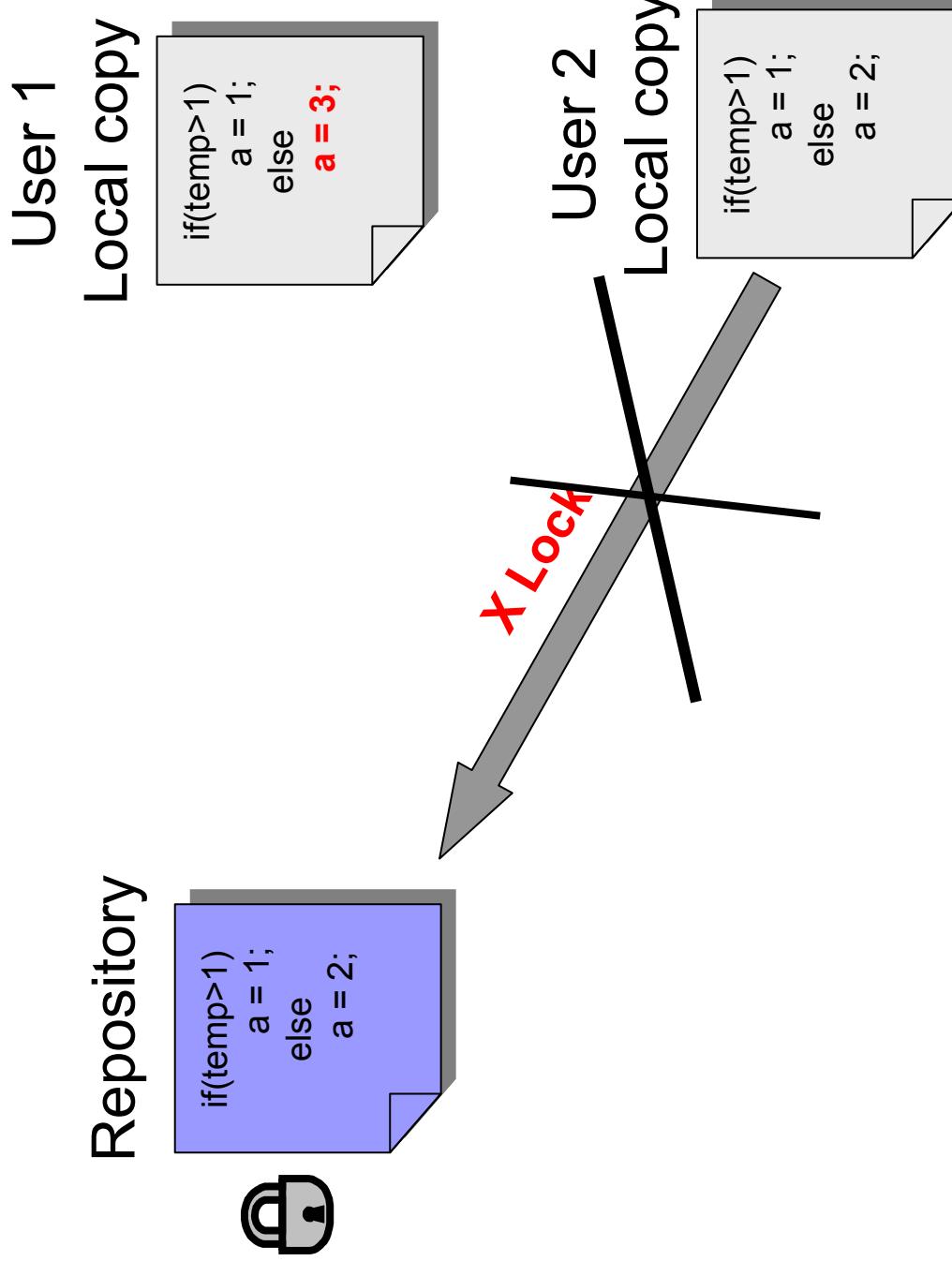
A Lock–Modify–Unlock megközelítés

- Módosítás előtt le kell lockolni egy file-t.
- Tehát egyszerre csak egy ember tudja módosítani a file-t. Olvasni tudja más is.

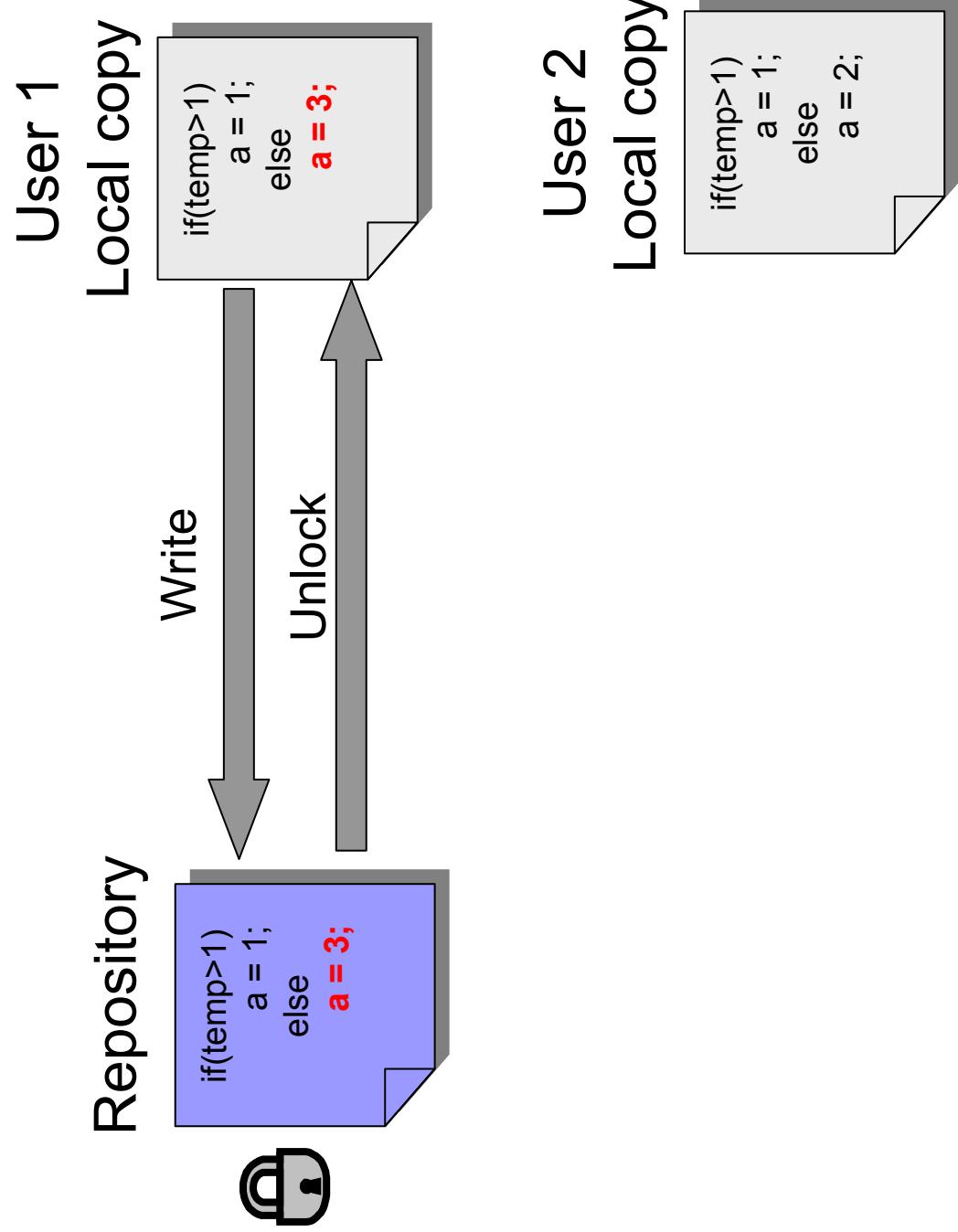
Lock–Modify–Unlock megközelítés



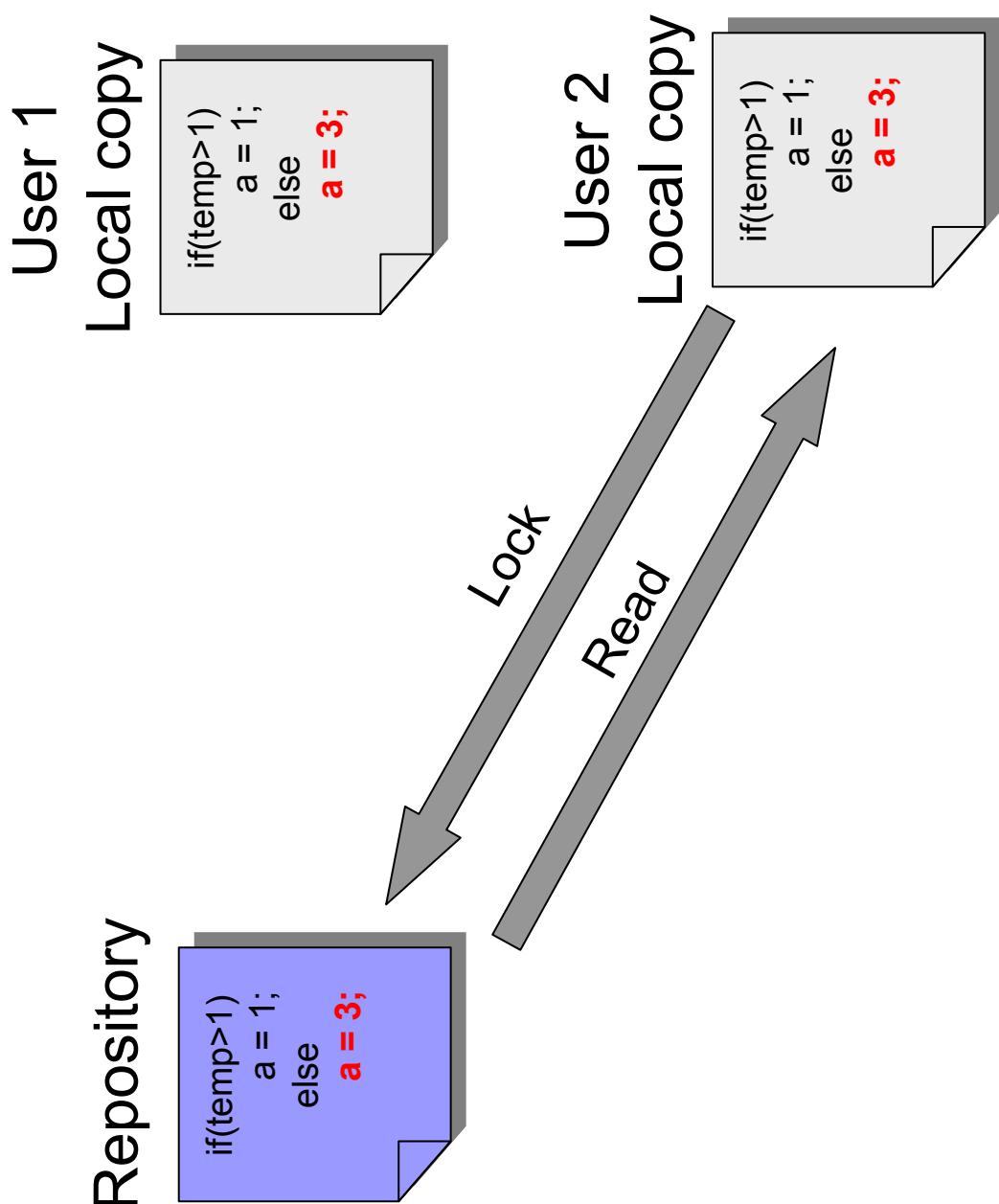
Lock–Modify–Unlock megközelítés

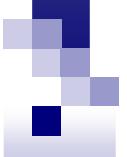


Lock - Modify - Unlock megközelítés



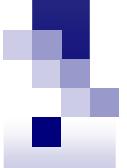
Lock - Modify - Unlock megközelítés





A Lock–Modify–Unlock megközelítés problémái

- Adminisztratív problémákhoz vezethet:
 - Ha egy fejlesztő elfelejt kilockolni egy file-t, akkor más nem ferhet hozzá.
 - Ha szabadságra megy, akkor pl. rendszergazda kell.
- Felesleges egymásra várást okozhat.
 - Egy C file-on belül például valaki az F1 függvényt akarja módosítani, másvalaki pedig az F2-t. Semmi köze a kettőnek egymáshoz mégsem tudják egyszerre megsinálni.
- A biztonság hamis illúzióját keltheti.
 - Például két fejlesztő dolgozik ugyanazon a projecten, az egyik lockolja az A file-t, a másik a B file-t. A két file között függőség áll fent. Mindkettőn azt hiszik biztonságban vannak, holott mégsem.



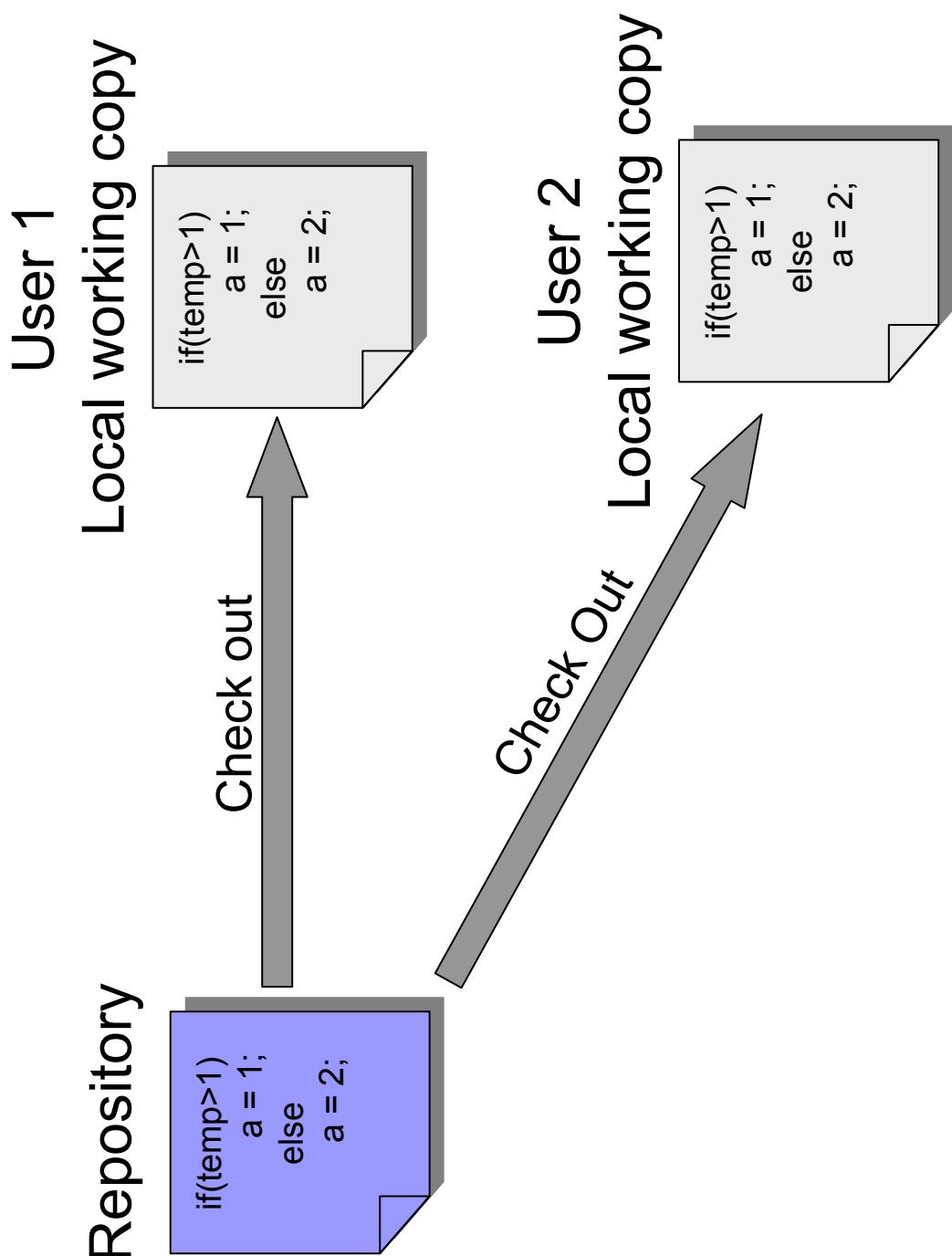
A Copy–Modify–Merge megközelítés

(CVS, Subversion stb. által használt módszer)

- Egyszerre több fejlesztő is ki „check out”-olhatja ugyanazt, mindenki a saját Working copy-ját használja.
- Working copy: A Repository (vagy annak egy részének) saját gépen található leképezése.
- A létrejövő konfliktusokat pedig Merge-gel, tehát fuzionálással oldják fel, és így hoznak létre egy új verziót.
- A Merge, bár támogatva van a verziókövető rendszer által, alapvetően mégis emberi döntésekkel követel, tehát nem automatikusan történik.

A Copy–Modify–Merge megközelítés

Mindketten ki „check out”-olják a file-t



A Copy–Modify–Merge megközelítés

Mindketten módosítják

Repository

```
if(temp>1)
    a = 1;
else
    a = 2;
```

User 1

Local working copy

```
if(temp>1)
a = 10;
else
    a = 2;
```

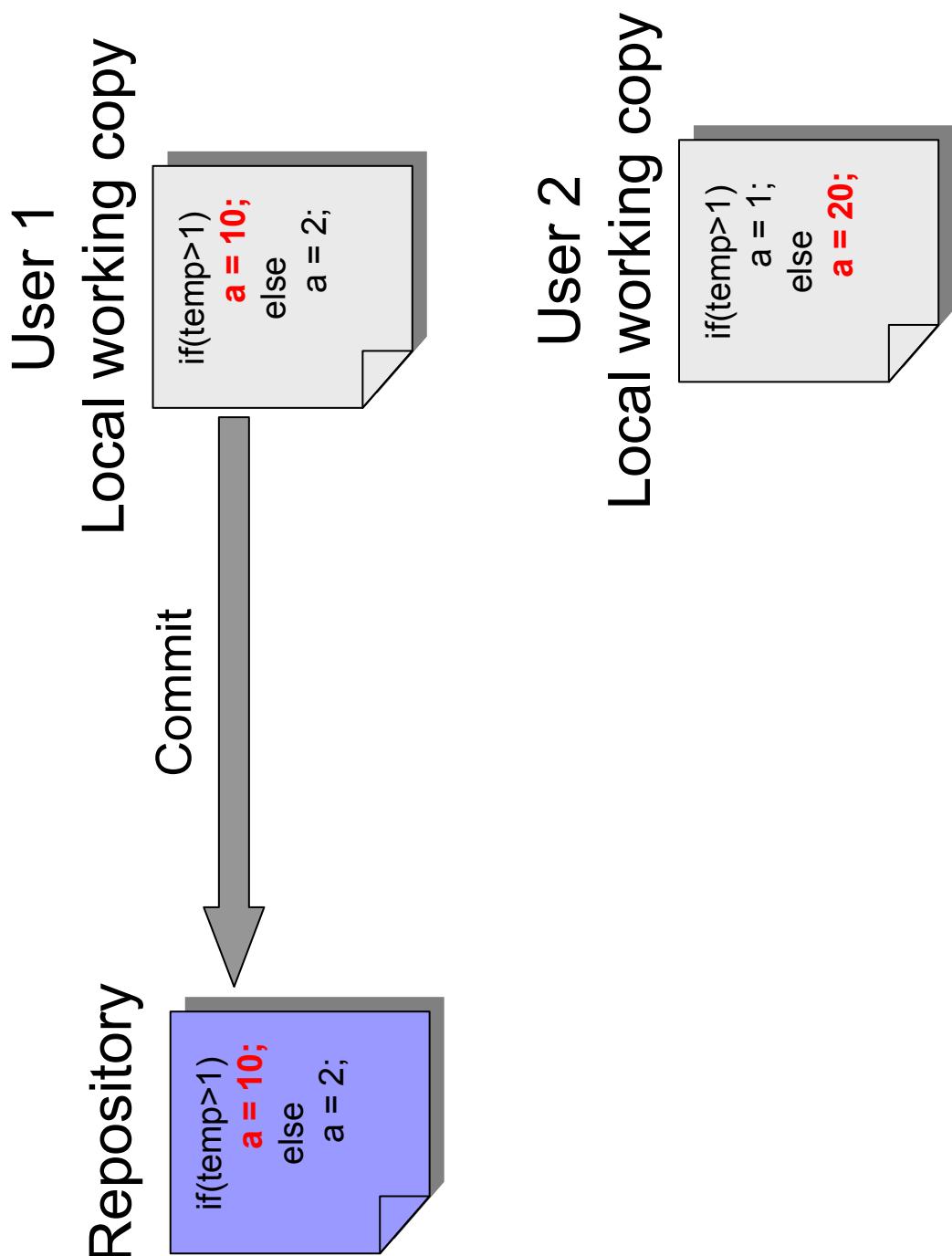
User 2

Local working copy

```
if(temp>1)
    a = 1;
else
a = 20;
```

A Copy–Modify–Merge megközelítés

User 1 végzett, feltölti a módosításokat



A Copy–Modify–Merge megközelítés

User 2 nem tudja feltölteni a módosításokat, mert az Ő local working copy-ja nem up-to-date

User 1

Repository

```
if(temp>1)  
    a = 10;  
else  
    a = 2;
```

Local working copy

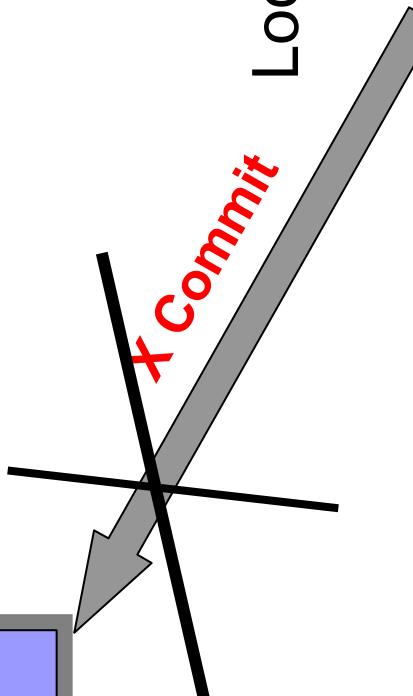
```
if(temp>1)  
    a = 10;  
else  
    a = 2;
```

User 2

```
if(temp>1)  
    a = 1;  
else  
    a = 20;
```

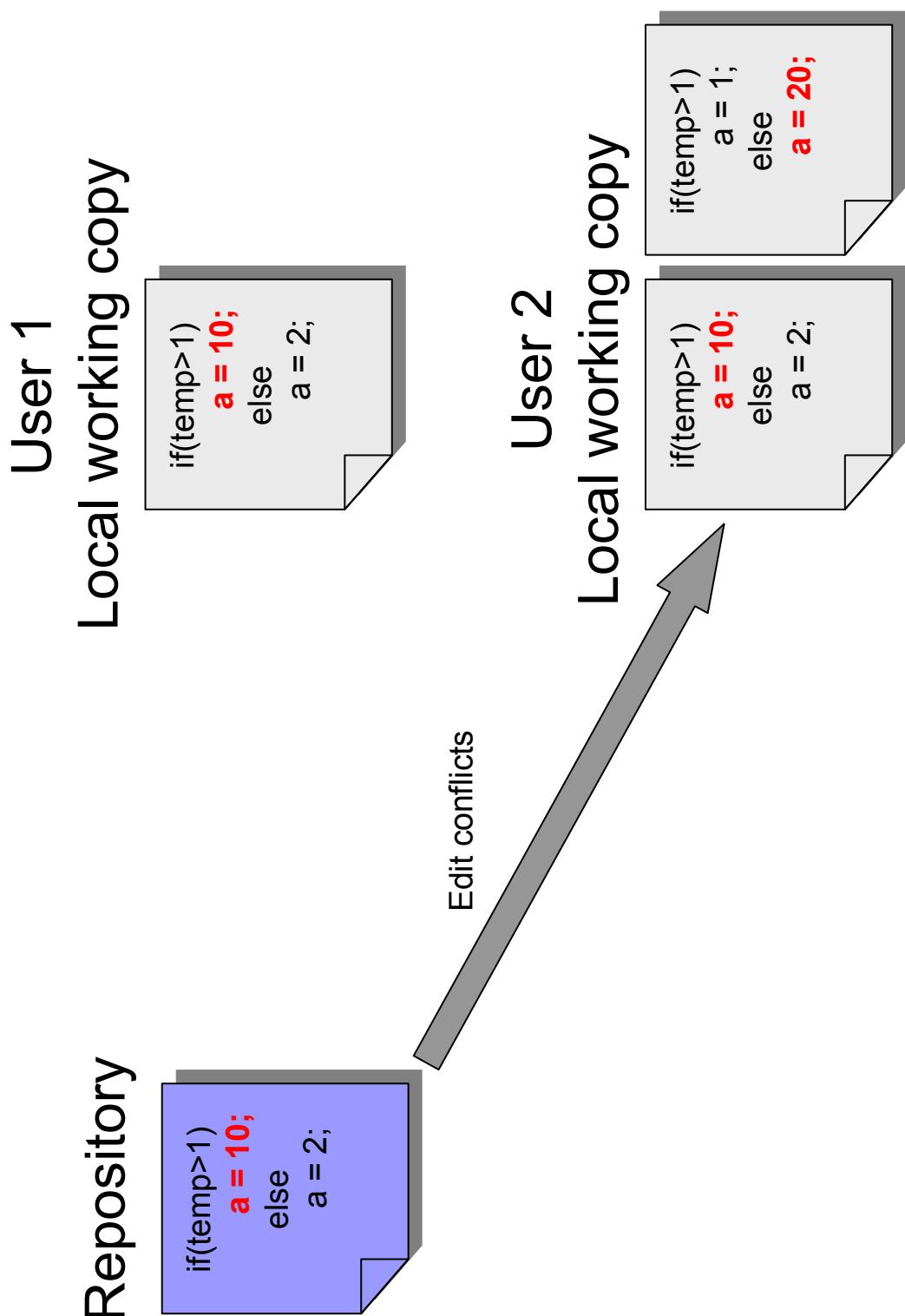
Local working copy

X Commit



A Copy–Modify–Merge megközelítés

User 2 kijelöli az új verziót



A Copy–Modify–Merge megközelítés

User 2 merges

Repository

```
if(temp>1)  
    a = 10;  
else  
    a = 2;
```

User 1

Local working copy

```
if(temp>1)  
    a = 10;  
else  
    a = 2;
```

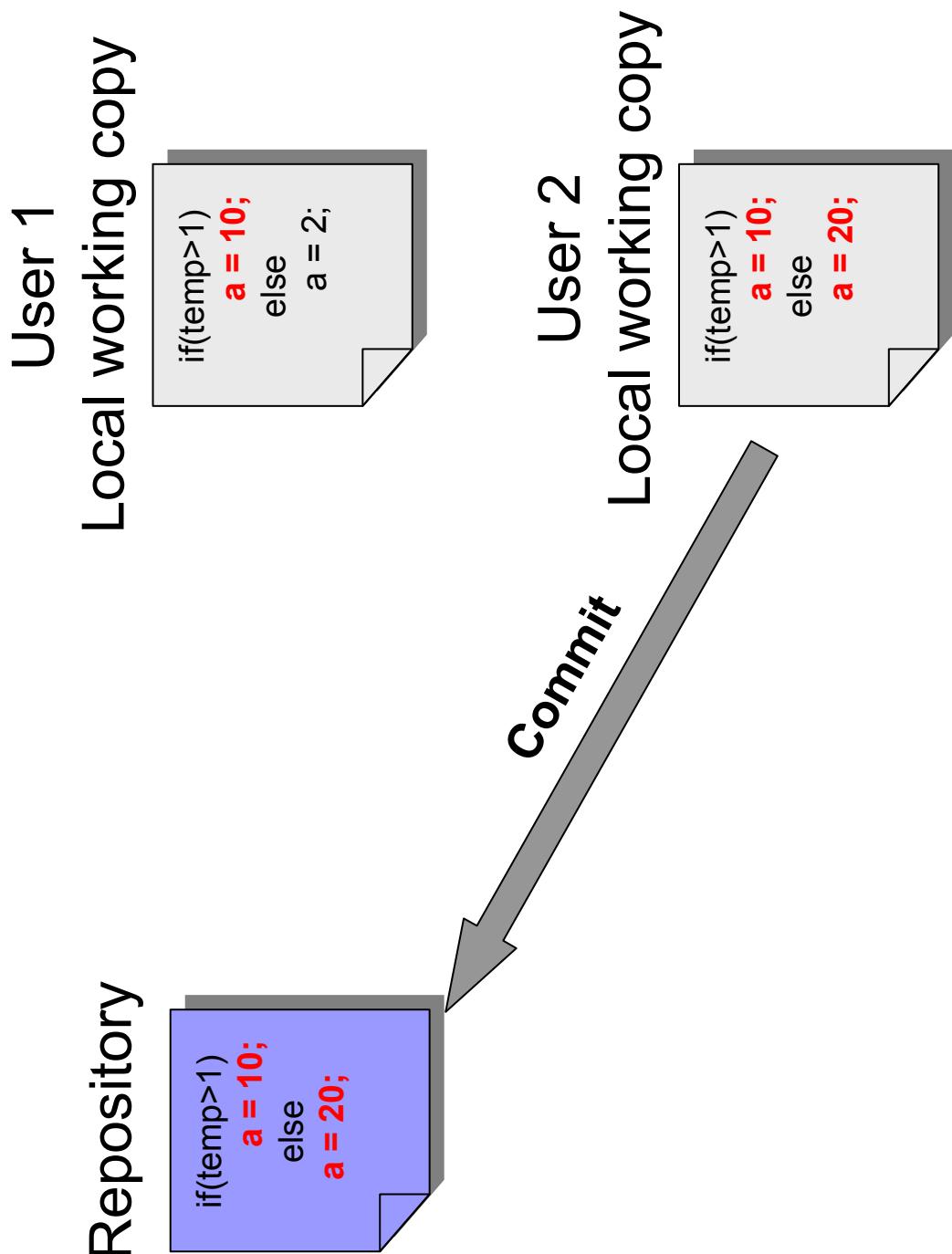
User 2

Local working copy

```
if(temp>1)  
    a = 10;  
else  
    a = 20;
```

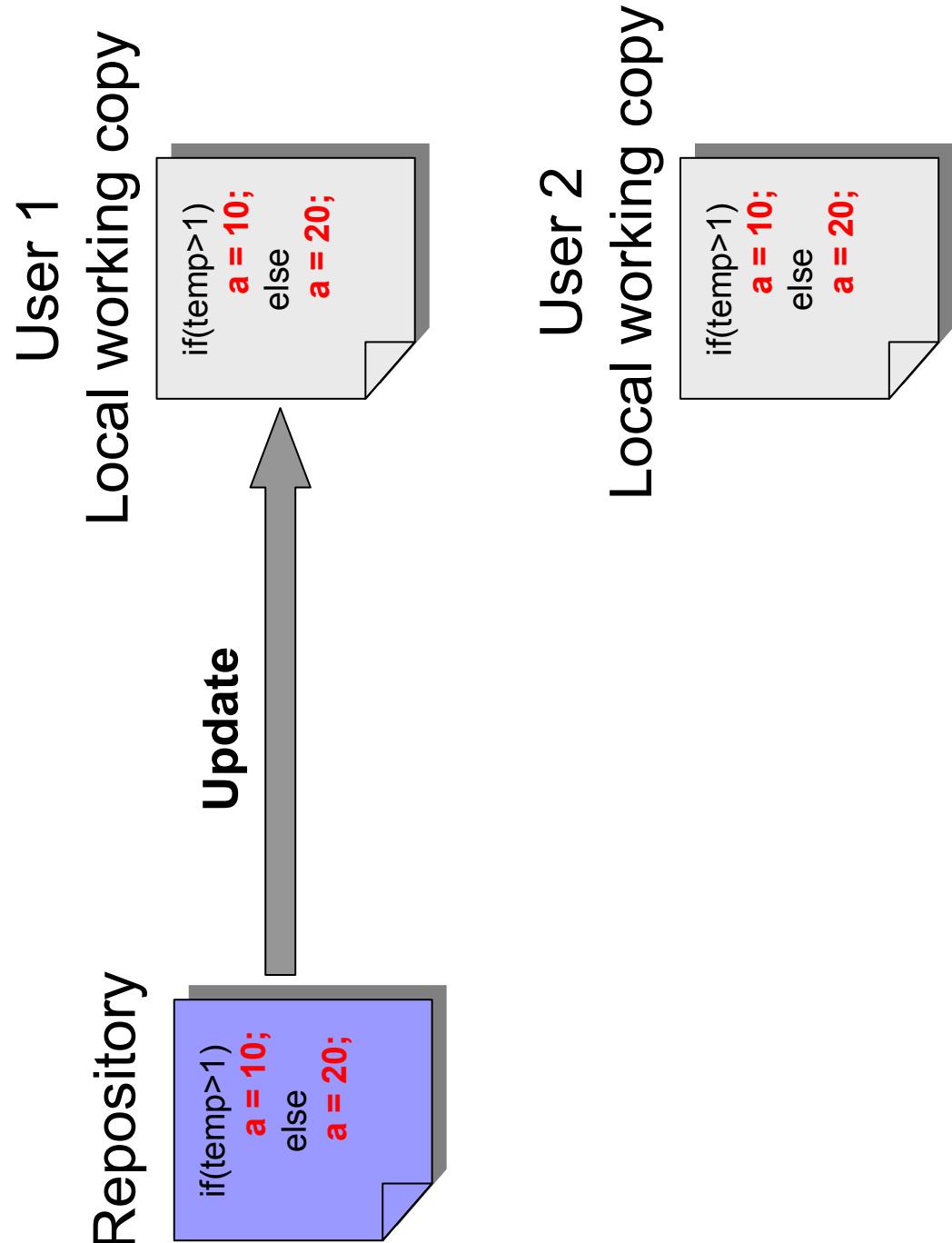
A Copy–Modify–Merge megközelítés

User 2 feltölti az új verziót



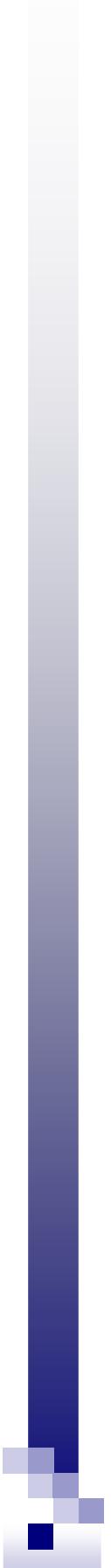
A Copy–Modify–Merge megközelítés

User 1 Update-el



A Copy–Modify–Merge megközelítés

- Egyszerre több fejlesztő is dolgozhat ugyanazon a kódon.
- Commit-nél az esetleges konfliktusok kiderülnek.
- Emberek kell döntenie a konfliktus feloldásáról.
- A verziókötő rendszer nem helyettesíti az emberek közötti kommunikációt.

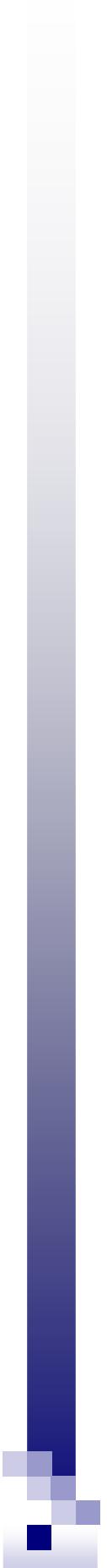


Melyek azok az esetek, amikor mégis lock-ot kell használni?

- Olyan bináris jellegű file-ok esetében, ahol a merge nem megoldható.
 - Például, hangfile-ok, bináris file-ok
 - NYÁK-tervezek, kapcsolási rajzok
- Ezért a legtöbb verziókövető rendszerben megtartották a lock funkciót.

Subversion (SVN)

- A Copy–Modify–Merge módszert használja.
- Open source
 - <http://subversion.tigris.org/>
- Eredetileg a CVS leváltására , egy „jobb CVS”.
- Apache webszerver alá telepíthető, alapvetően a szerver, létezik windows-os egyből települő változat is.
- Alapvetően command line kliens, amihez sok grafikus user interface változat létezik.



Subversion working copy-k

- Helyi gépen lévő másolata a repository-nak vagy annak egy darabjának.
- Tartalmaz még egy .svn könyvtárat könyvtáronként, ami tartalmazza, hogy melyik file került módosításra stb.
- minden commit művelet hatására egy snapshot jön létre a repository-ról.

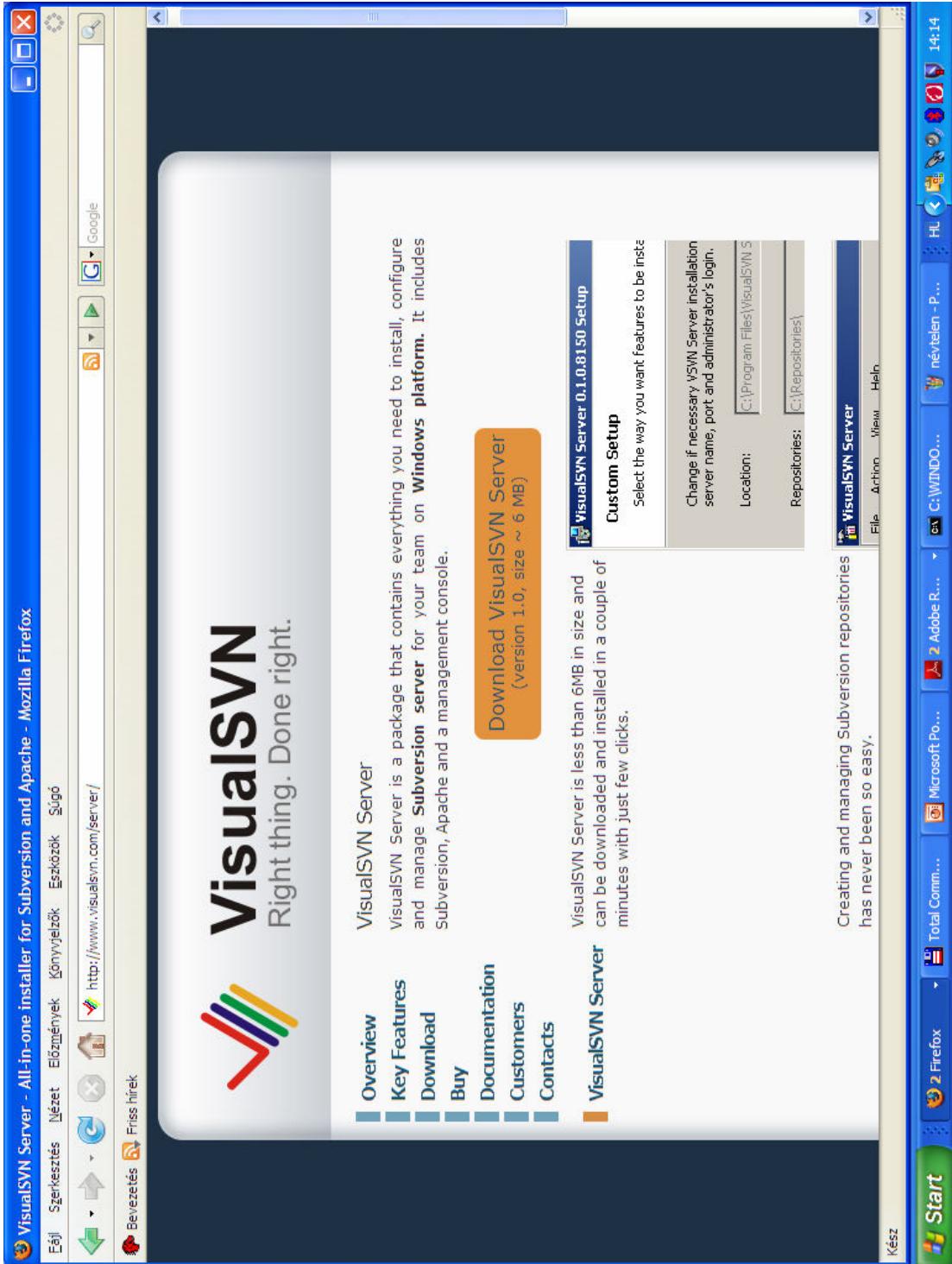
Subversion használata

Első lépések: server telepítése

- Apache webszerver alá
- Általában nem a mi dolgunk ezt telepíteni és beállítani a különböző felhasználókat.
- Létezik Visual SVN telepítő, amely egyszerűsíti a problémát.

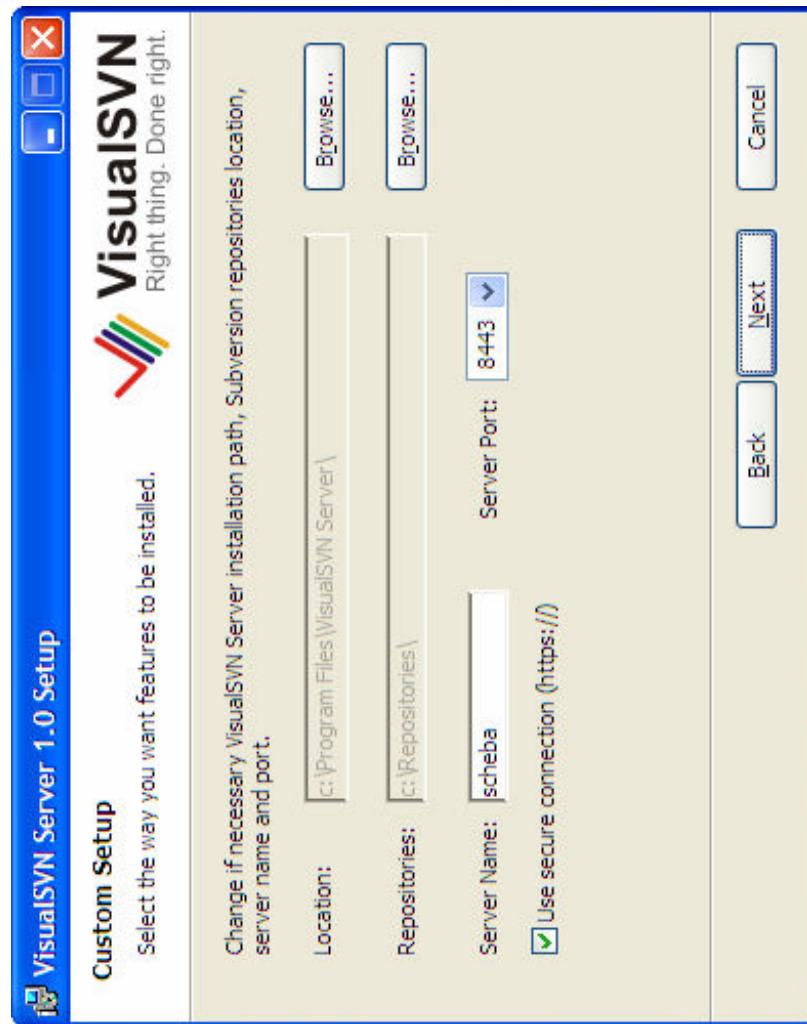
Subversion server Windows-on

<http://www.visualsvn.com/server>



VisualSVN (telepítés)

- Gyakorlatilag next nyomogatás, a szerver nevet kell megadni.



Repository és annak elemei

Második lépés: a repository létrehozása

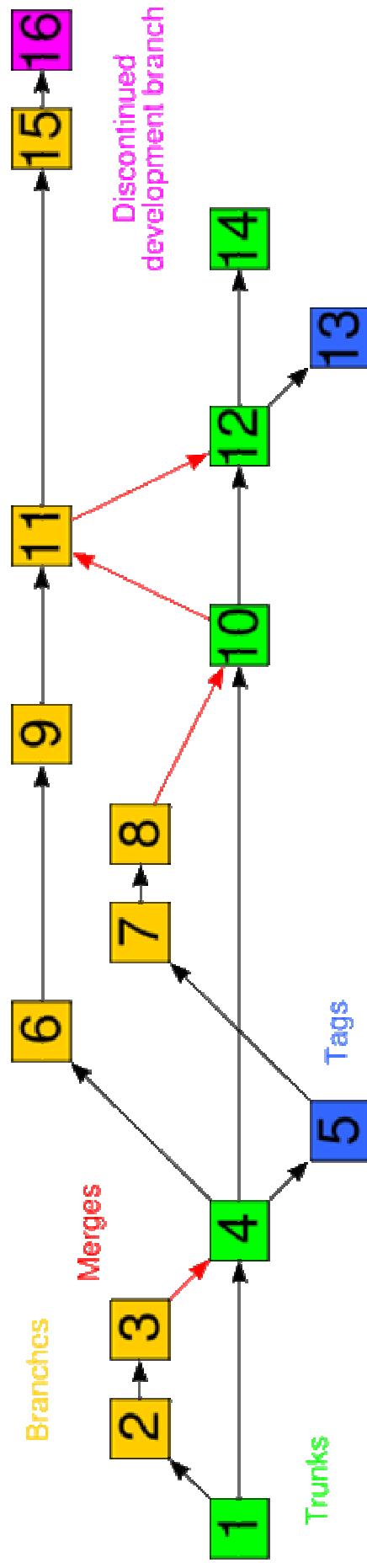
- Repository (raktár): *Központi nyilvántartása az adatoknak vagy projectnek (a master copy).*
- A legtöbb repository projectenként a következő alapkönyvtárat tartalmazza:
 - trunk
 - Main line of development
 - tags
 - Release-ek, stabil verziók
 - branches
 - Az elágazások követésére
- minden commit művelet hatására egy snapshot jön létre a repository-ról.
- Az összes a repository-ban nyilvántartott elemhez (file, vagy könyvtár) tartozik egy ún. Revision number.
 - A Revision number egy integer szám, amely minden változtatásnál inkrementálódik .

Repository változása egy project során

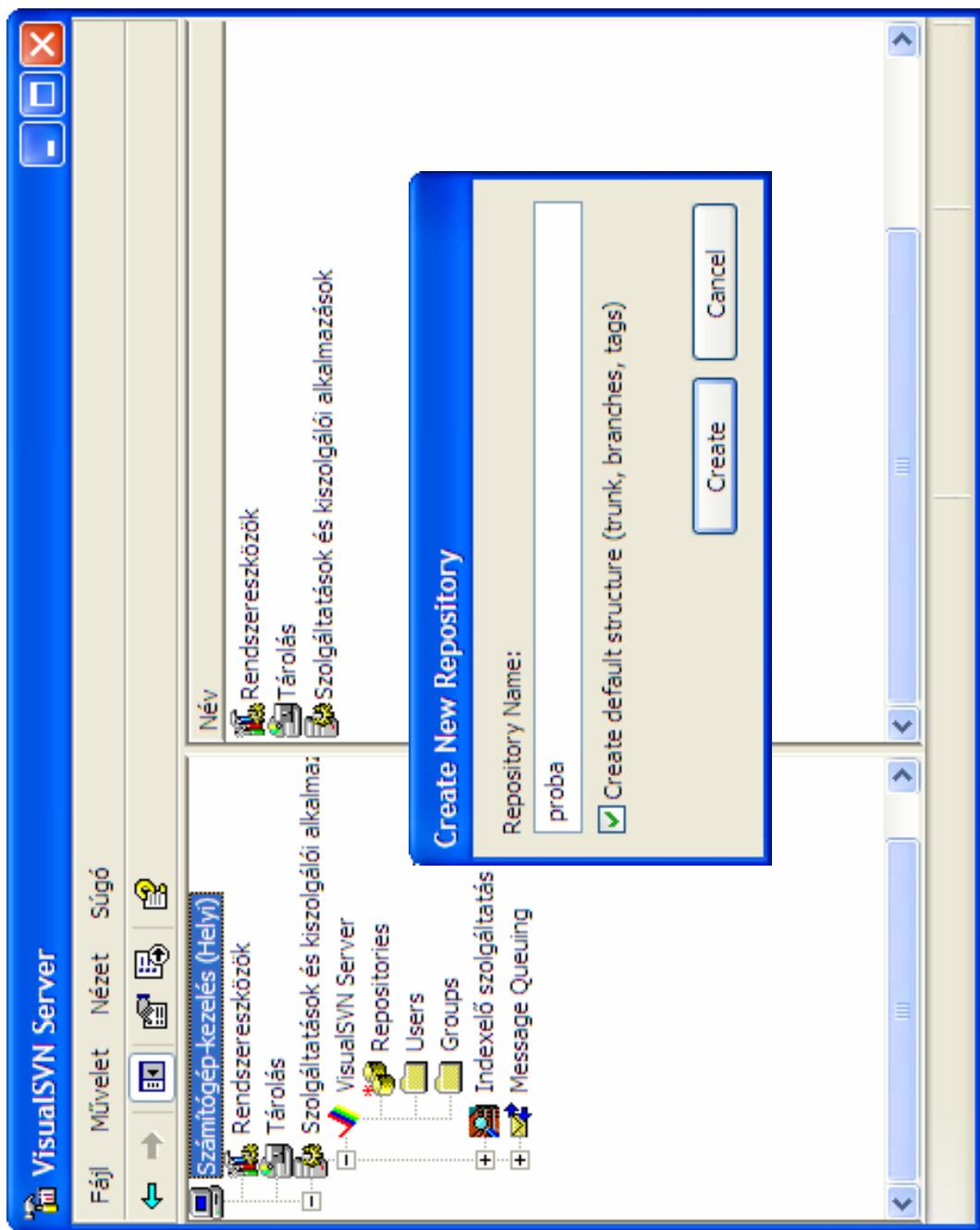
Második lépés: a repository létrehozása

- A legtöbb repository projectenként a következő alap könyvtárakat tartalmazza

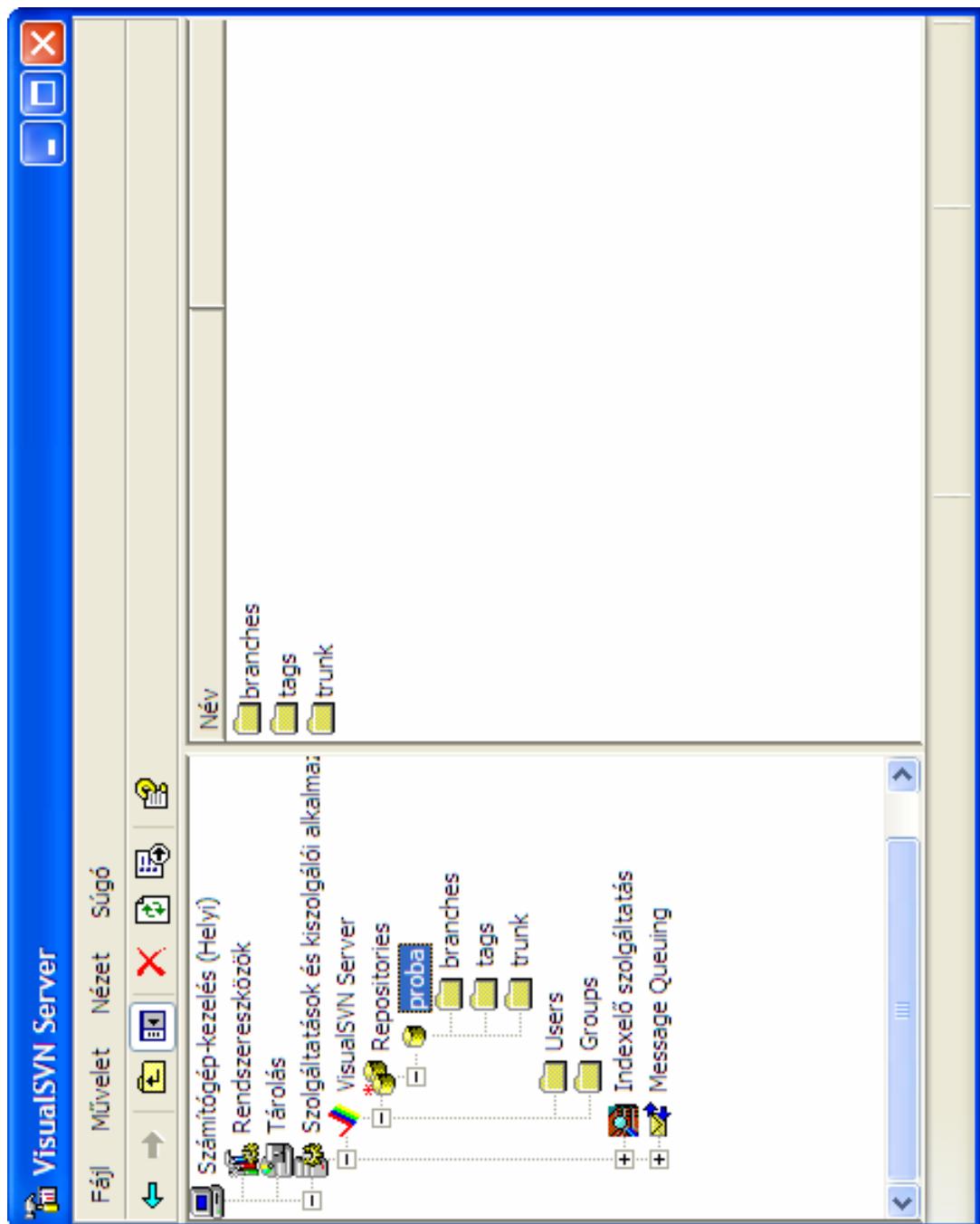
- Trunk: Main line of development
- tags: Release-ek, stabil verziók
- Branches: Az elágazások követésére



VisualSVN (Repository létrehozása 1)

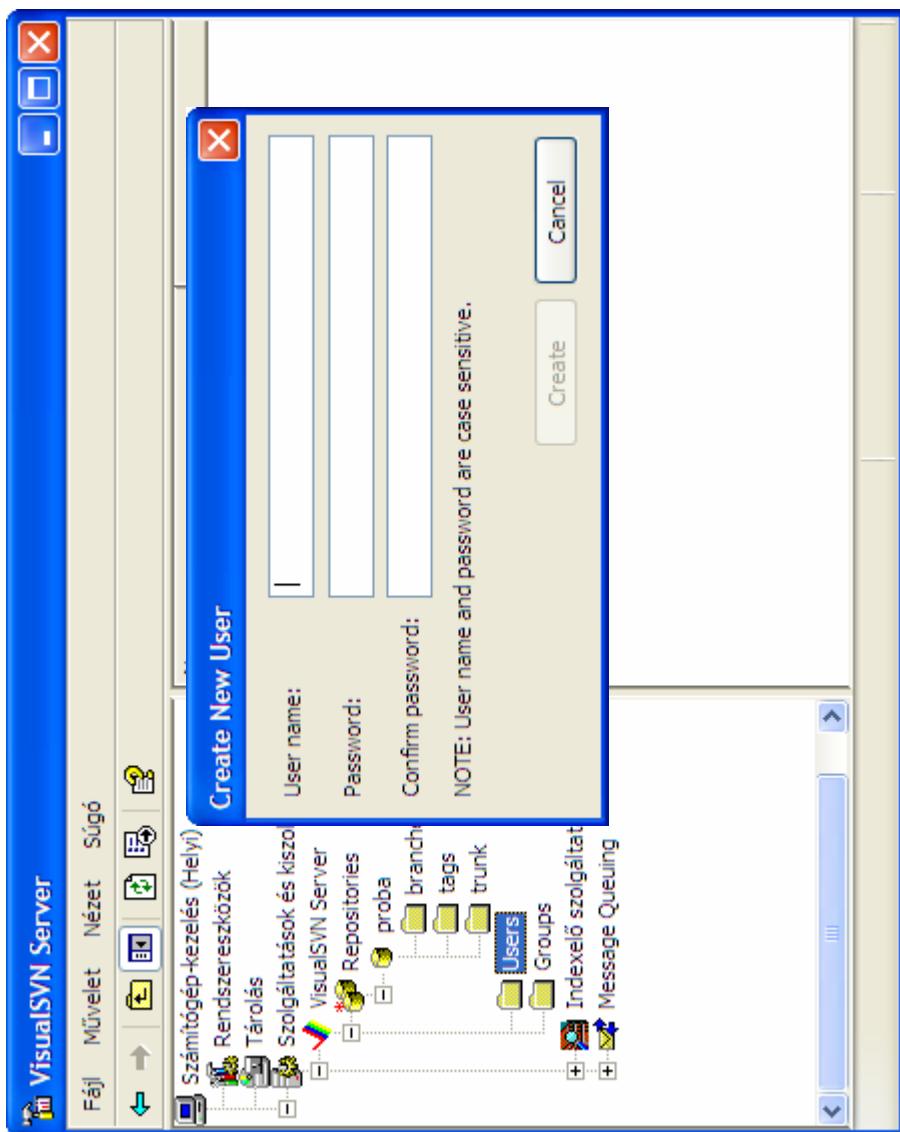


VisualSVN (Repository létrehozása 2)

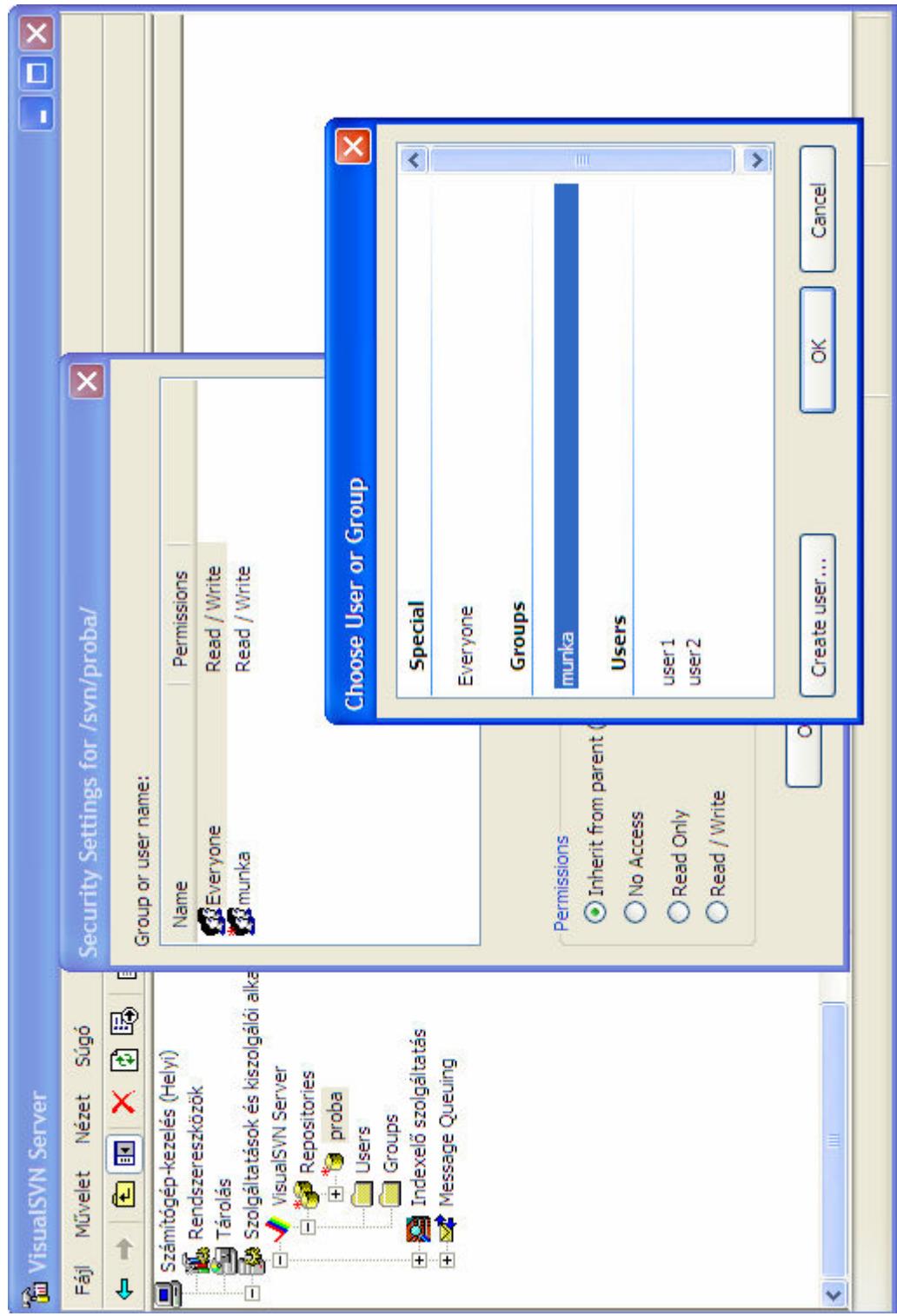


VisualSVN (User hozzáadása)

Harmadik lépés: a felhasználók hozzáadása

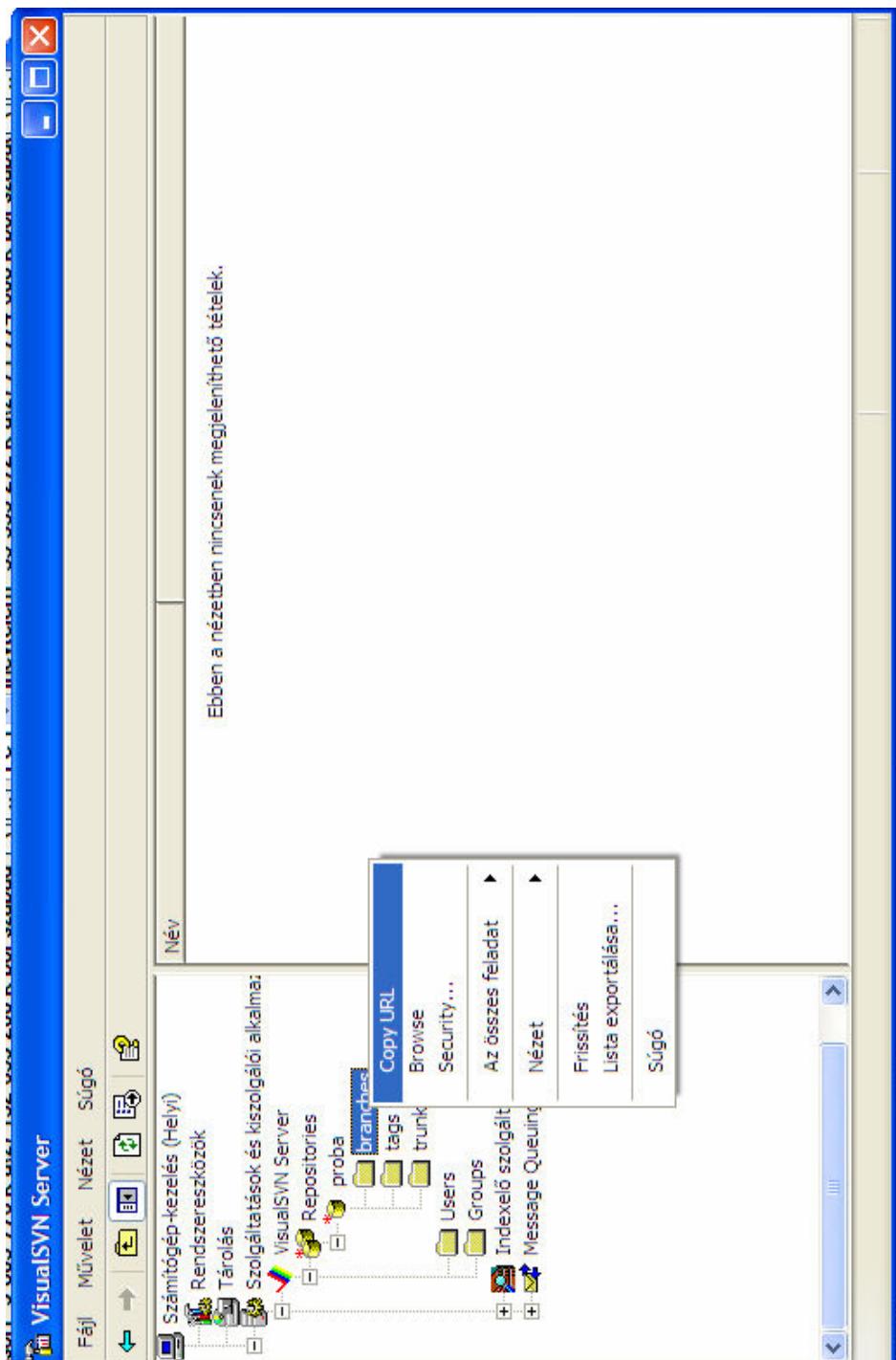


VisualSVN (User hozzárendelése egy repository-hoz)



VisualSVN (repository címének másolása)

Ezt kell majd megadni a kliensnek



Client Tortoise SVN

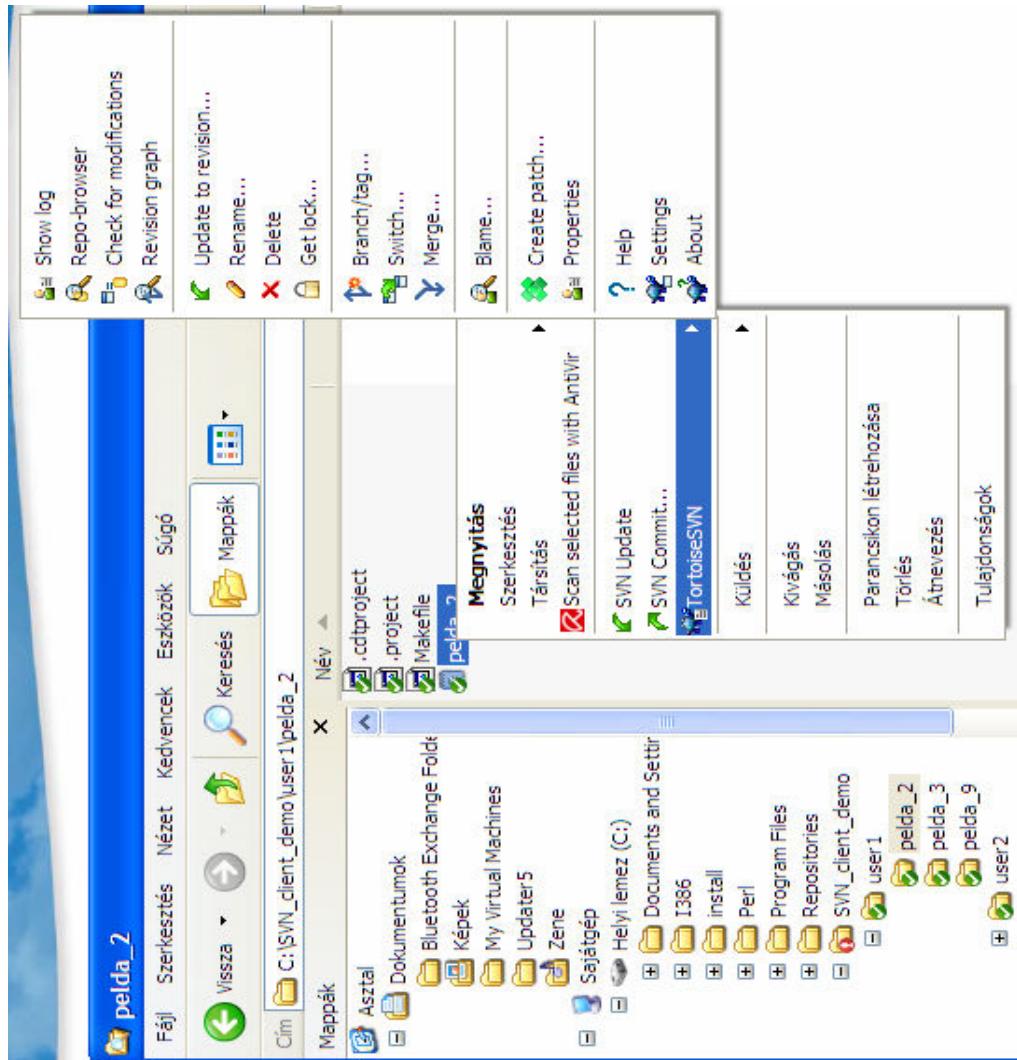
Negyedik lépés: kliens telepítése

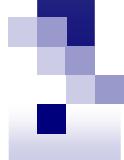
The screenshot shows a Mozilla Firefox browser window with the following details:

- Title Bar:** TortoiseSVN | The coolest Interface to (Sub)Version Control - Mozilla Firefox
- Address Bar:** http://tortoisessvn.net/
- Toolbar:** Back, Forward, Stop, Home, Refresh, Stop, Google search bar.
- Menu Bar:** Fájl, Szolgáltatás, Nézet, Előzmények, Könnyűelőzök, Eszközök, Súgó
- Content Area:**
 - TortoiseSVN Logo:** A blue cartoon tortoise wearing a white hard hat and holding a wrench.
 - Text:** TortoiseSVN The coolest Interface to (Sub)Version Control
 - Navigation Links:** About, Donate, Download, Project status, Get Help, Tools
 - Awards:** SOURCEFORGE.NET® 2007 COMMUNITY CHOICE AWARDS WINNER - PROJECT OF THE YEAR
 - Search Bar:** Search
 - Text:** The easiest way to use SCM / source control software for Microsoft Windows and maybe the best Subversion client there is. It is implemented as a Windows shell extension, which makes it integrate seamlessly into the Windows explorer. Since it's not an integration for a specific IDE you can use it with whatever development tools you like. Another advantage of this shell integration is that not only web or software developers can use it but all the people who don't have the development tools installed, for example translators, doc writers, ...
 - Ads by Google:** VisualSVN, Syncro SVN Client
 - Navigation:** About, Download, Help, Community, Docs
 - Footer:** Kész

TortoiseSVN

- Ingyenes SVN kliens
(létezik CVS változat is)
□ <http://tortoisessvn.net/>
- Windows-ba beépülő
felhasználói felület

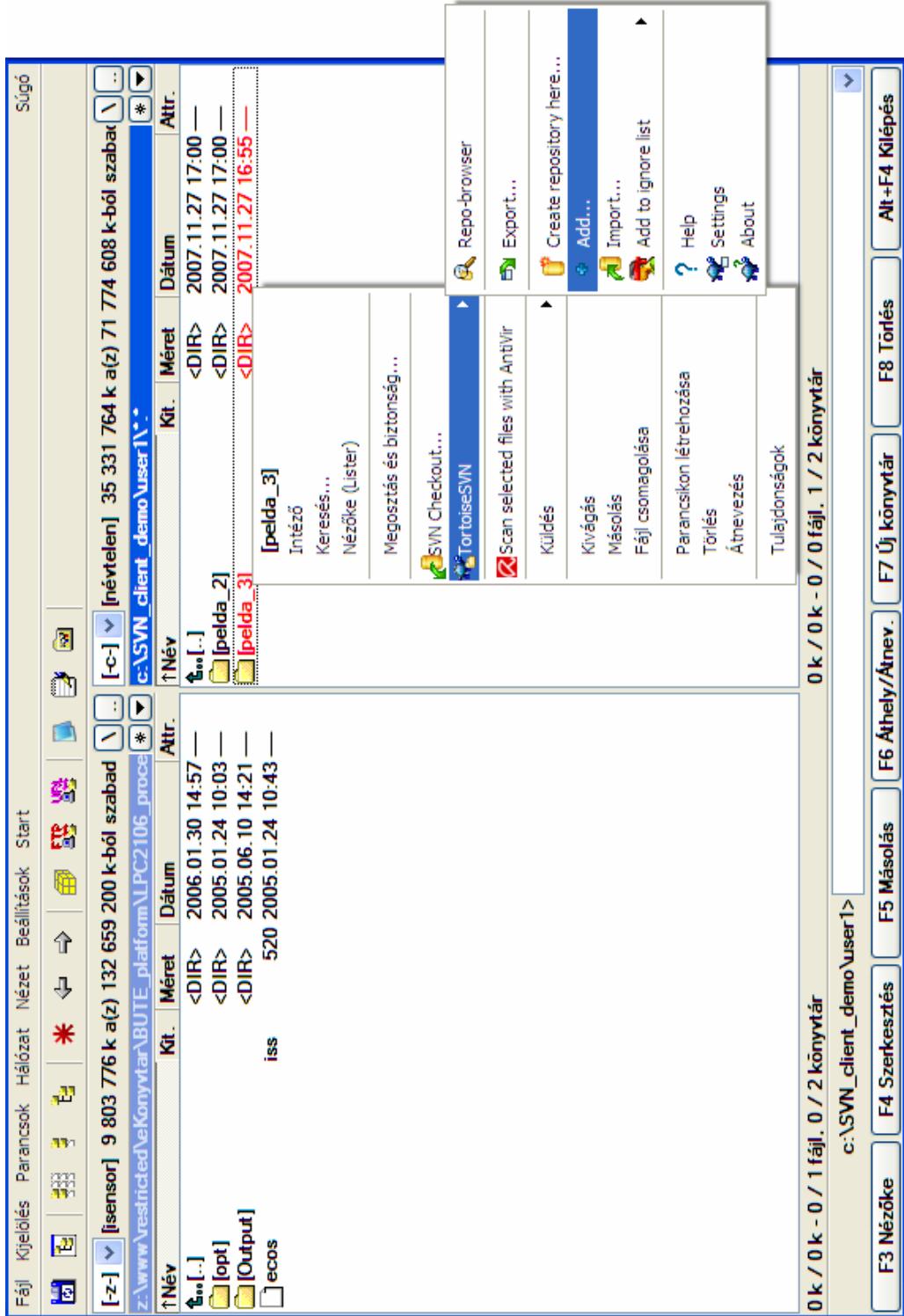




Hozzá adás az SVN rendszerhez

Ötödik lépés: új file-ok hozzáadása a verziókövetőhöz

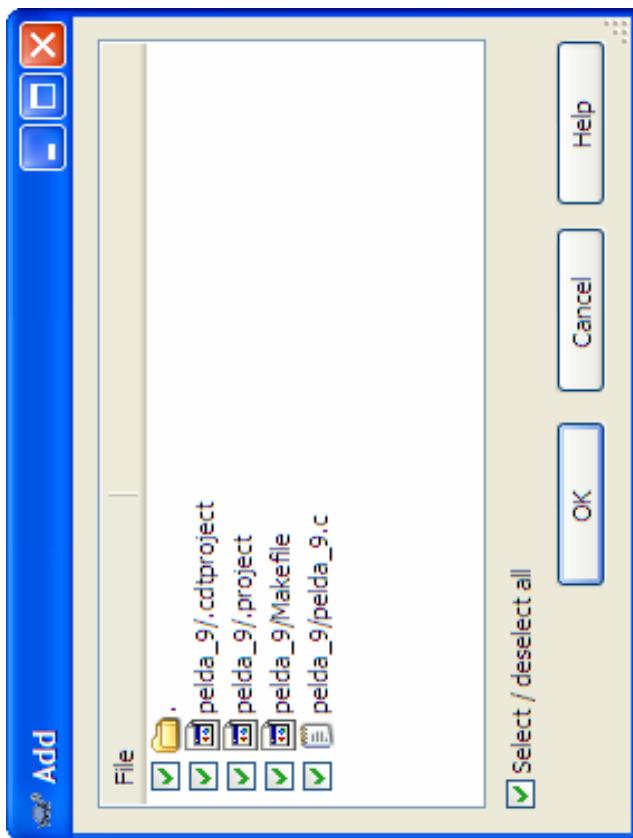
Add..



File-ok hozzáadása 2

Ki kell választanunk, hogy mely file-ok fognak szerepelni a rendszerben.

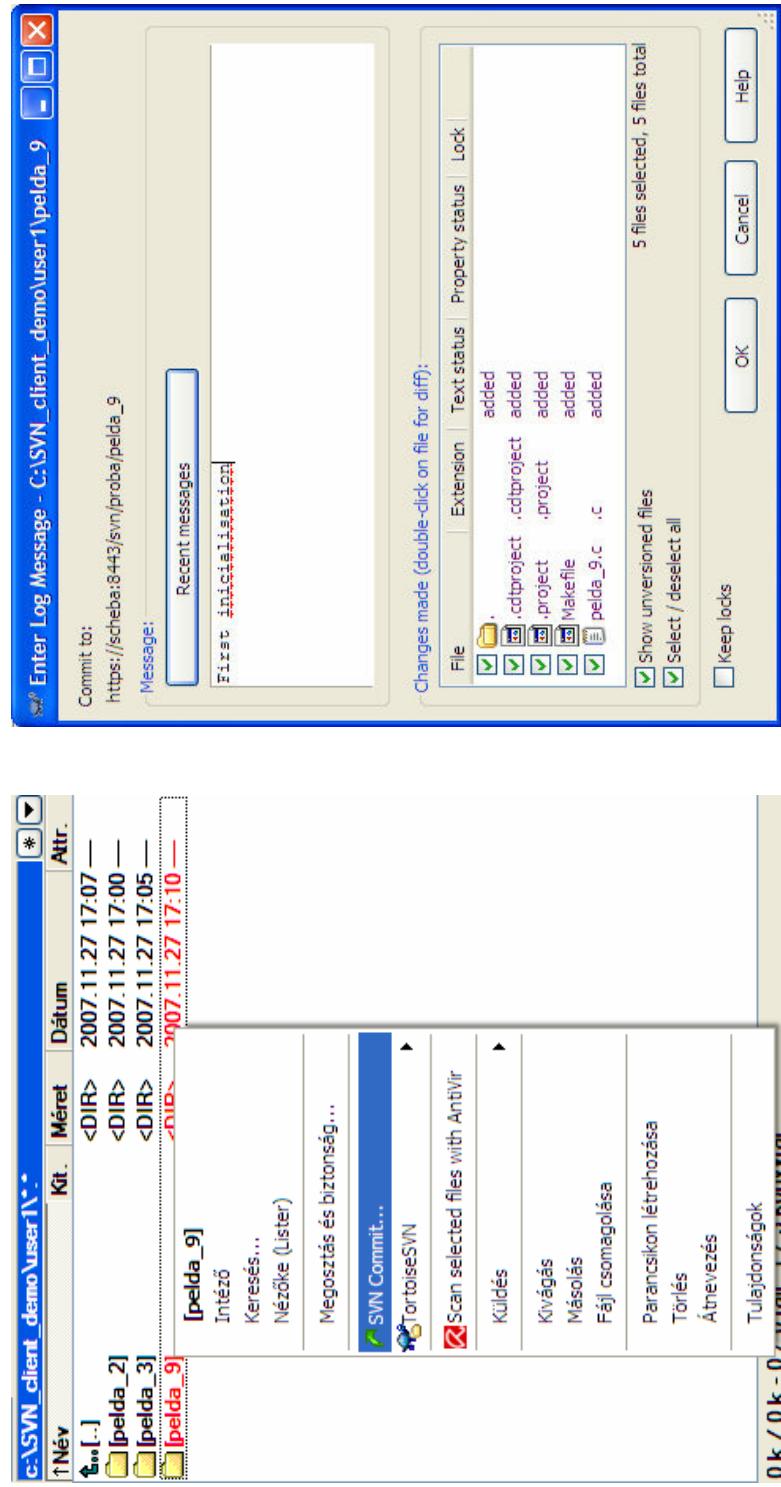
- minden hozzáadott könyvtárhoz létrejön a saját rejtett SVN könyvtár.
- nem célszerű olyan file-okat hozzáadni a rendszerhez, amelyek nem tartoznak szorosan a verziókövetés alá
 - Object file-ok
 - Hex file-ok is, bár ez alól van kivétel.



Commit...

Hatódik lépés: a repository első feltöltése

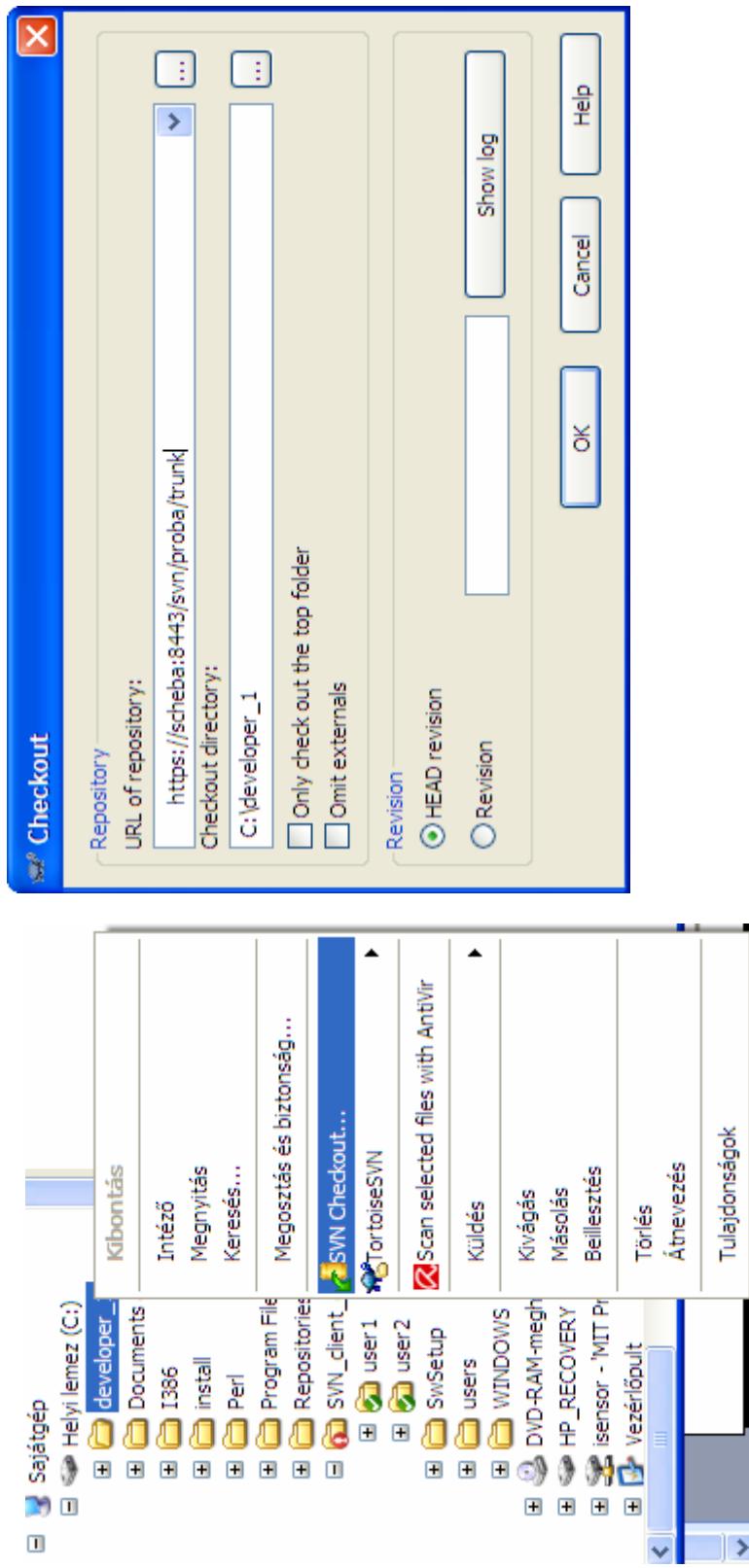
- Még nincsenek hozzáadva a repository-hoz, csak commit-elní lehet.
- Fontos, hogy minden új hozzáadáshoz egy note-ot kell írnunk.
 - Ez teszi lehetővé, hogy összefoglaljuk mit változtattunk.
 - A verziókötő rendszer nem varázsló, nem találja ki a gondolatunkat.



Új felhasználó hozzáérése a projecthez

Check out...

- Még nincs saját working copy-nk, létre kell hozni. Kijelöljük, hová akarjuk a saját sand boxunkat, és Check out...



Check out...

- Leszedi az új verziókat
- minden up-to-date

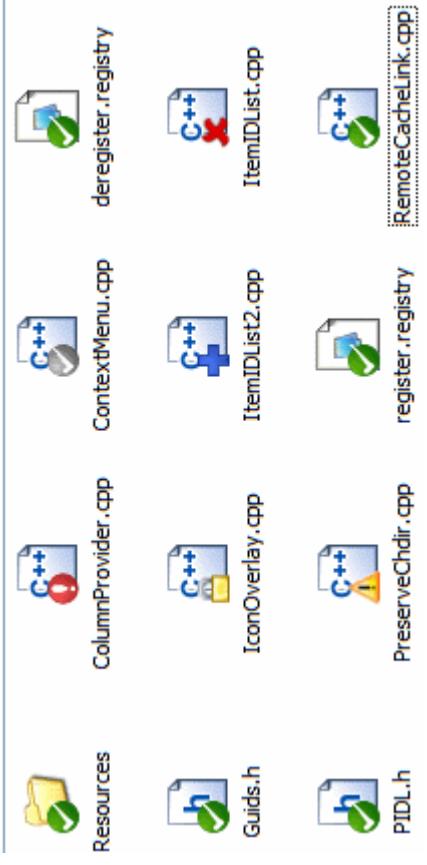
The screenshot shows two windows related to a SVN checkout operation.

TortoiseSVN Checkout Dialog: This window lists the files being added during the checkout. The log shows 15 additions, totaling 32 kB transferred. The path listed is C:\developer_1\pelda_2.

Action	Path	Mime type
Added	C:\developer_1\pelda_2	
Added	C:\developer_1\pelda_2\cdtproject	
Added	C:\developer_1\pelda_2\project	
Added	C:\developer_1\pelda_2\c	
Added	C:\developer_1\pelda_2\Makefile	
Added	C:\developer_1\pelda_3	
Added	C:\developer_1\pelda_3\cdtproject	
Added	C:\developer_1\pelda_3\project	
Added	C:\developer_1\pelda_3\c	
Added	C:\developer_1\pelda_3\Makefile	
Added	C:\developer_1\pelda_9	
Added	C:\developer_1\pelda_9\cdtproject	
Added	C:\developer_1\pelda_9\project	
Added	C:\developer_1\pelda_9\c	
Added	C:\developer_1\pelda_9\Makefile	

Tortoise File Browser: This window shows the local directory structure after the checkout. It includes a 'Mapákk' tab where the checked-out files are listed. The files include .cdtproject, CDPROJECT.fájl, Makefile, and several pelda_x files.

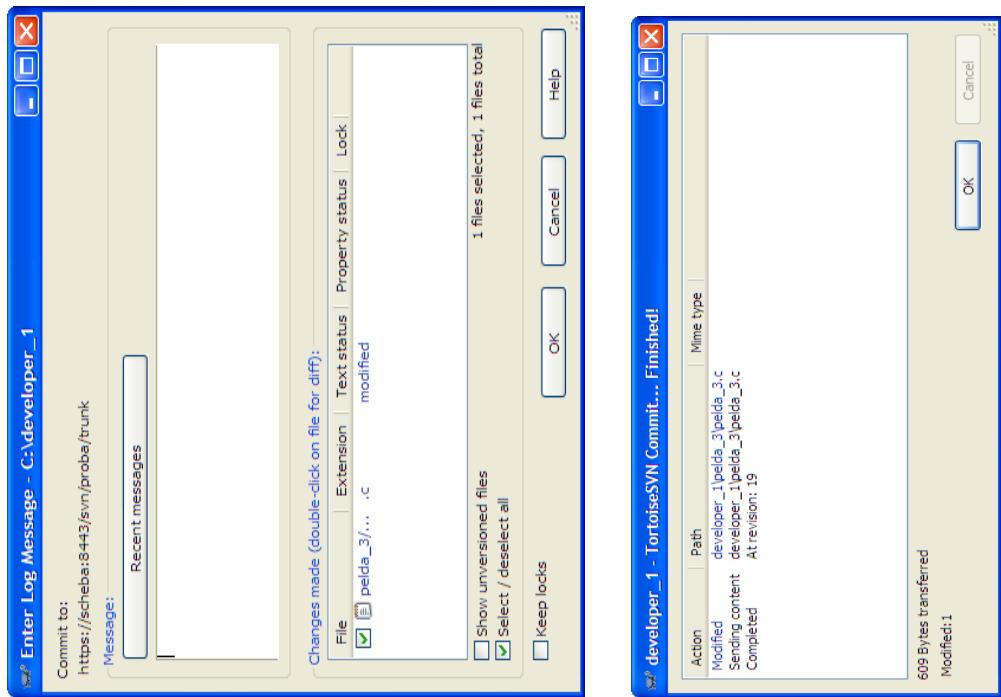
Ikon magyarázatok



- Normal: stáatus, up-to-date
- Modified: helyi módosítás
- ⚠ Conflict: nem updatelehető
- Read-only valaki lockol-ta, amíg nem tudjuk lockolni, addig csak olvashatjuk.
- 🔒 Locked: Zárolva
- ✖ Deleted: Törölve
- ✚ Added: Új hozzáadva

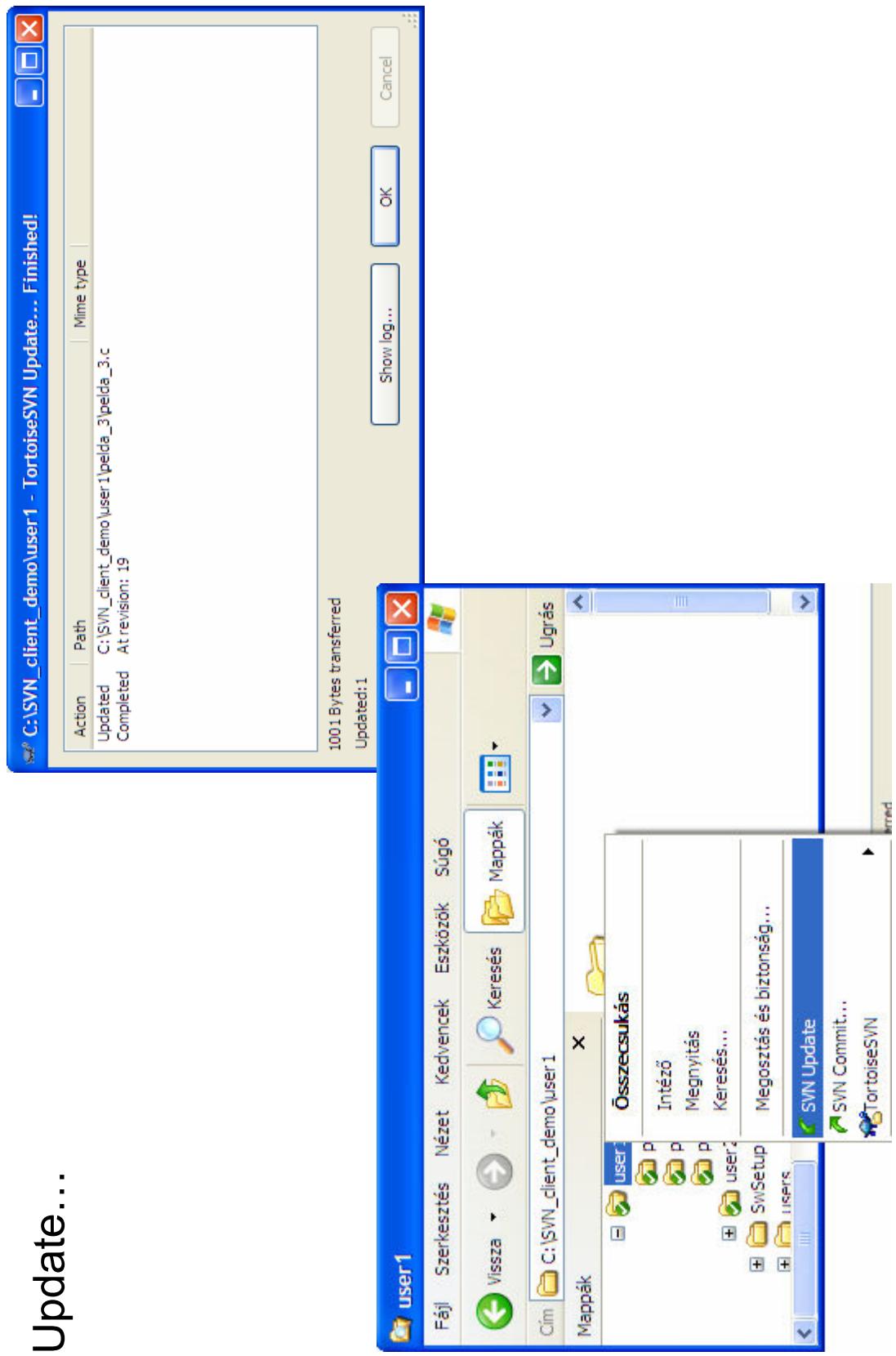
Helyi módosítás és annak felvitele

- Commit...
 - Miért nem jó ez így?
 - Meg kellene nézni, hogy volt-e változtatás a projecten (egyébként szólt volna).
 - Össze kellene venni a változásokat.



Többiek követik a módosítást

- Update...



De mi volt a módosítás?

- Update...

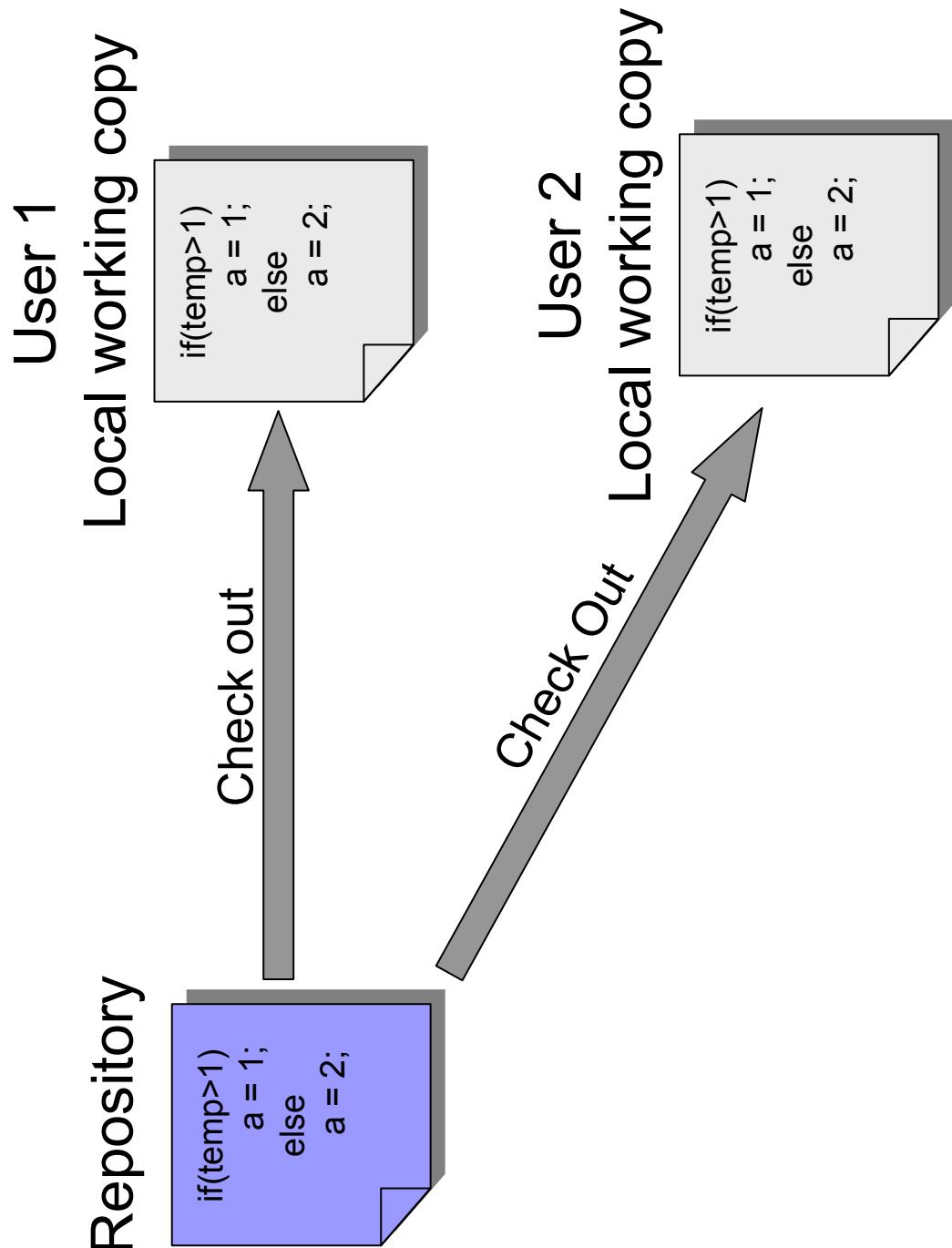
The screenshot shows two windows from the TortoiseSVN interface. The top window is titled 'C:\developer_1 - TortoiseSVN Update... Finished!' and displays a summary of the update: Action: Updated, Path: C:\Developer\1\pelida_3\pelida_3.c, Mime type: Completed, At revision: 20, Updated: 1, 1001 Bytes transferred. It includes buttons for 'Compare with working copy', 'Show log...', 'OK', and 'Cancel'. The bottom window is titled 'TortoiseMerge' and shows a file comparison between 'pelida_3.c Revision 18' and 'pelida_3.c Revision 19'. The code editor displays the following C code:

```
28 *while(1)
29 {
30     diag_printf("Thread-A: %s\r\n");
31     cyg_thread_delay(50);
32 }
33
34 }
35
36
37 /*
38 The·Thread·B
39 */
40 void·B_thread(cyg_addrword_t index)
41 {
42
43
44     while(1)
45     {
46         diag_printf("Thread-B: %s\r\n");
47         cyg_thread_delay(100);
48     }
49 }
```

The code editor has three panes: Left View (-2), Right View (+2), and Conflicts: 0. A status bar at the bottom right indicates 'For Help, press F1. Scroll horizontally with Ctrl+Scrollwheel'.

Több user módosít

Mindketten ki „check out”-olják a file-t



Mindketten módosítják

Repository

```
if(temp>1)
    a = 1;
else
    a = 2;
```

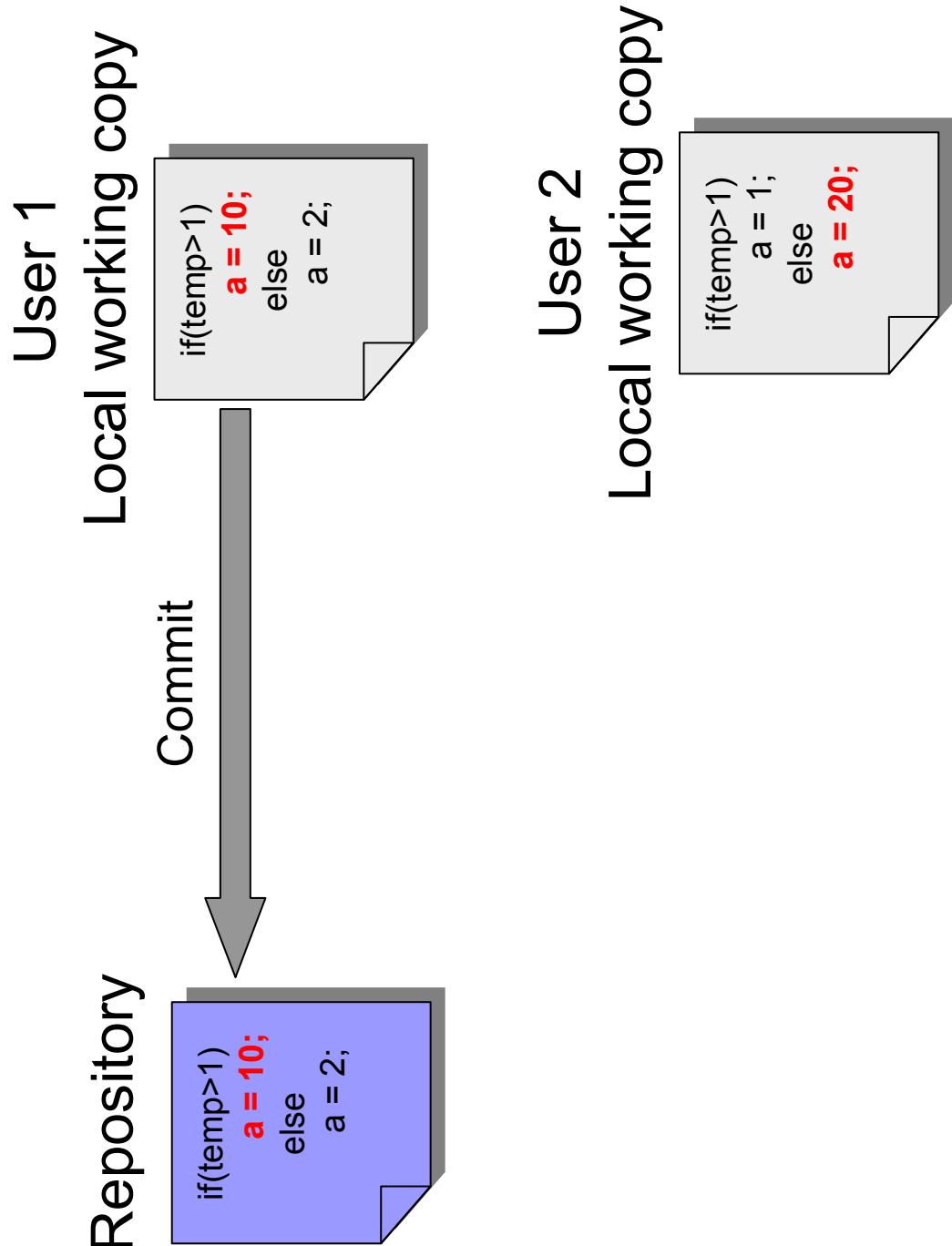
User 1 Local working copy

```
if(temp>1)
a = 10;
else
    a = 2;
```

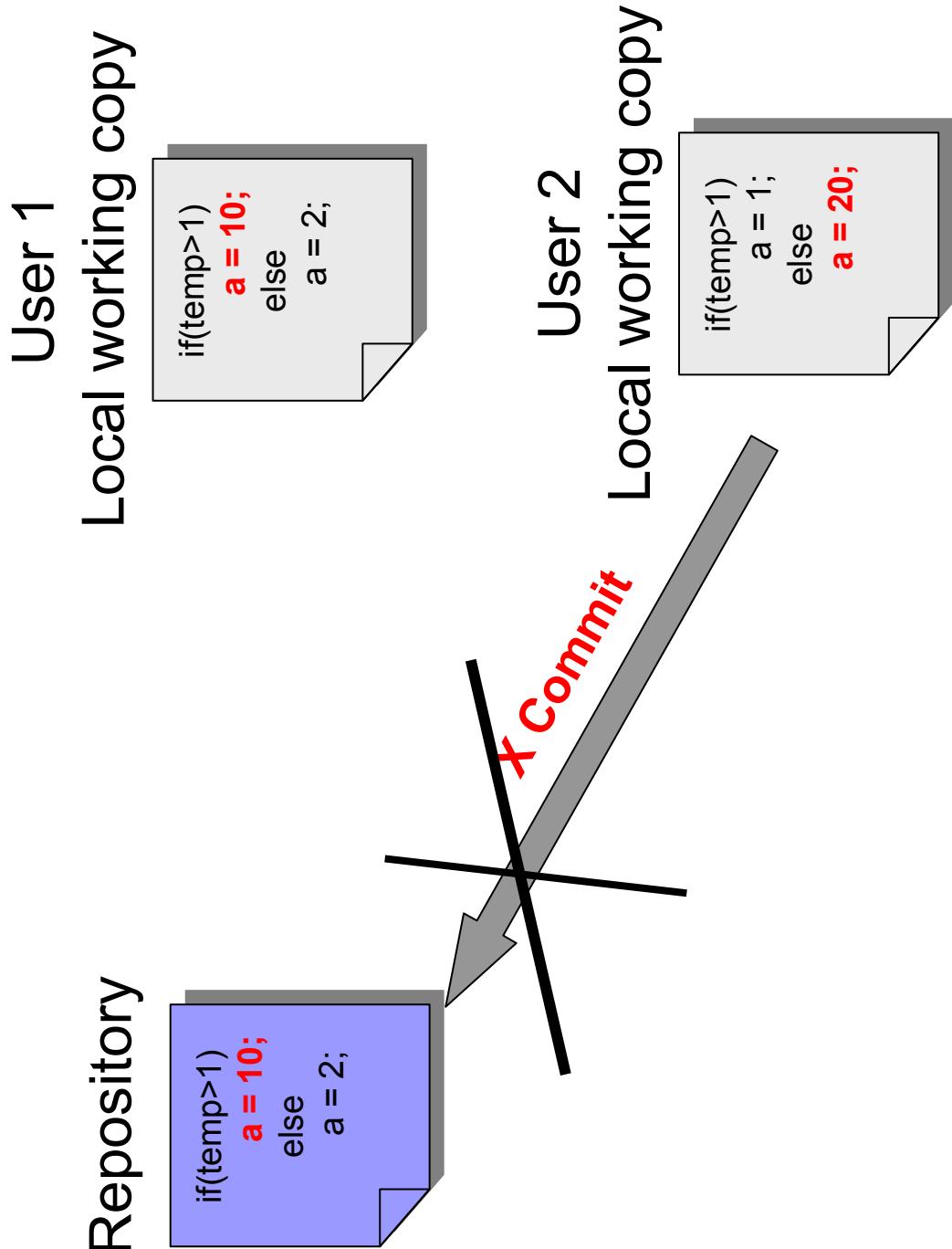
User 2 Local working copy

```
if(temp>1)
    a = 1;
else
a = 20;
```

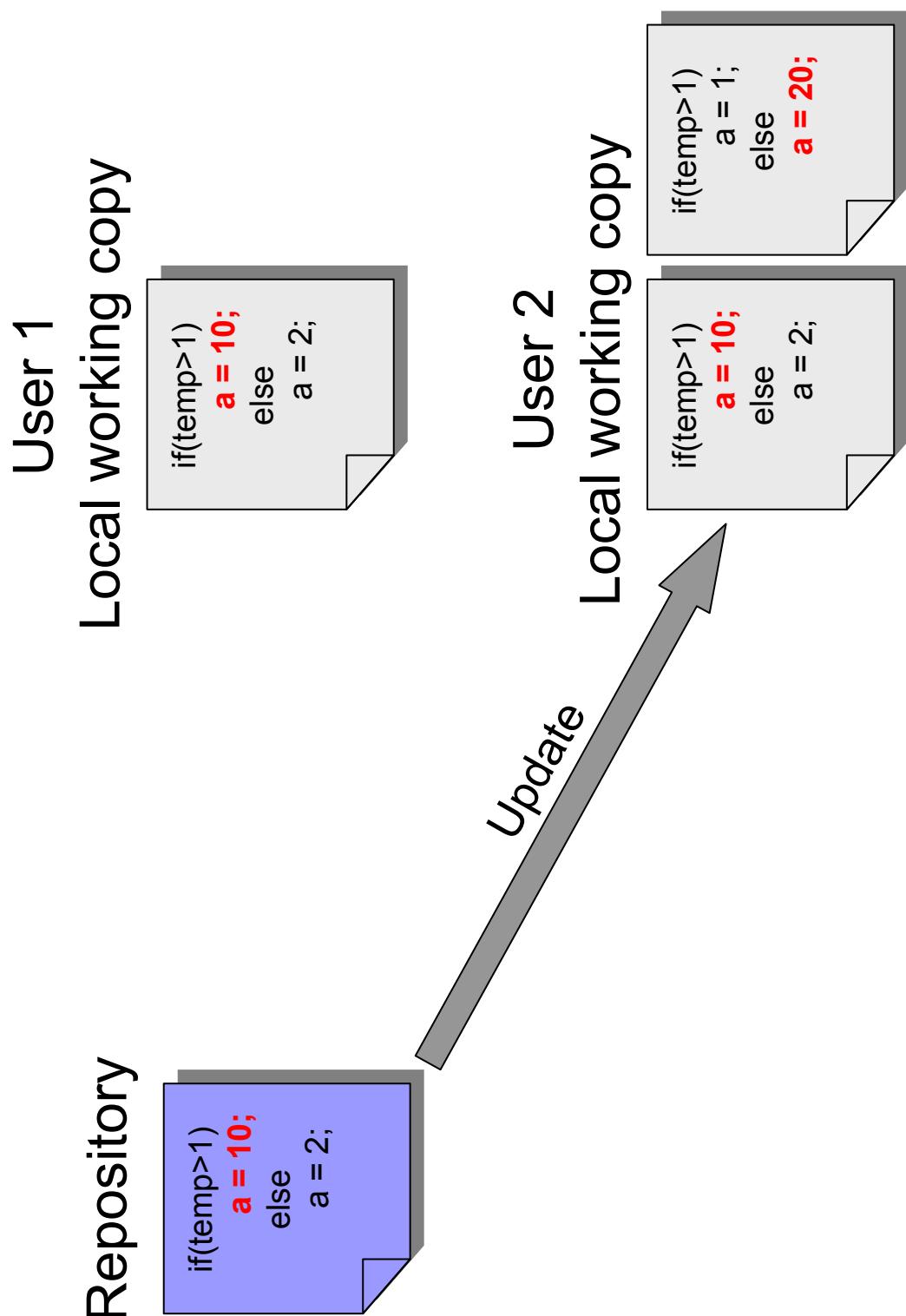
User 1 végzett, feltölti a módosításokat



User 2 nem tudja feltölteni a módosításokat, mert az ö local working copy-ja nem up-to-date



User 2 kiolvassa az új verziót



User 2 merges

Repository

```
if(temp>1)  
    a = 10;  
else  
    a = 2;
```

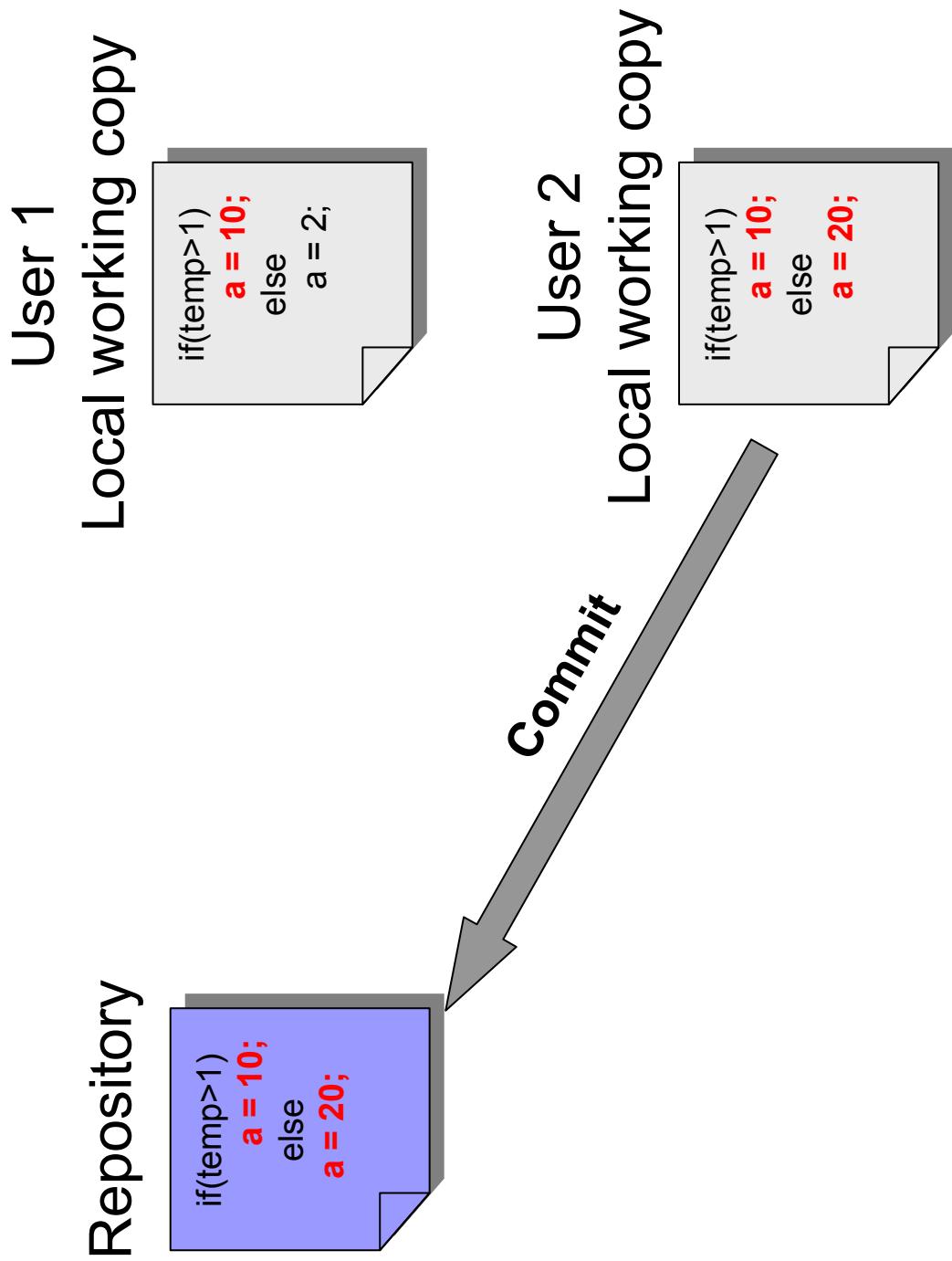
User 1 Local working copy

```
if(temp>1)  
    a = 10;  
else  
    a = 2;
```

User 2 Local working copy

```
if(temp>1)  
    a = 10;  
else  
    a = 20;
```

User 2 feltölti az új verziót



Több user módosít (Elso lépések: Check..)

- Az egyik user javít valamit, és feltölti
- A másik is módosít, és mit kell tennie

- Első lépések: Check for modification

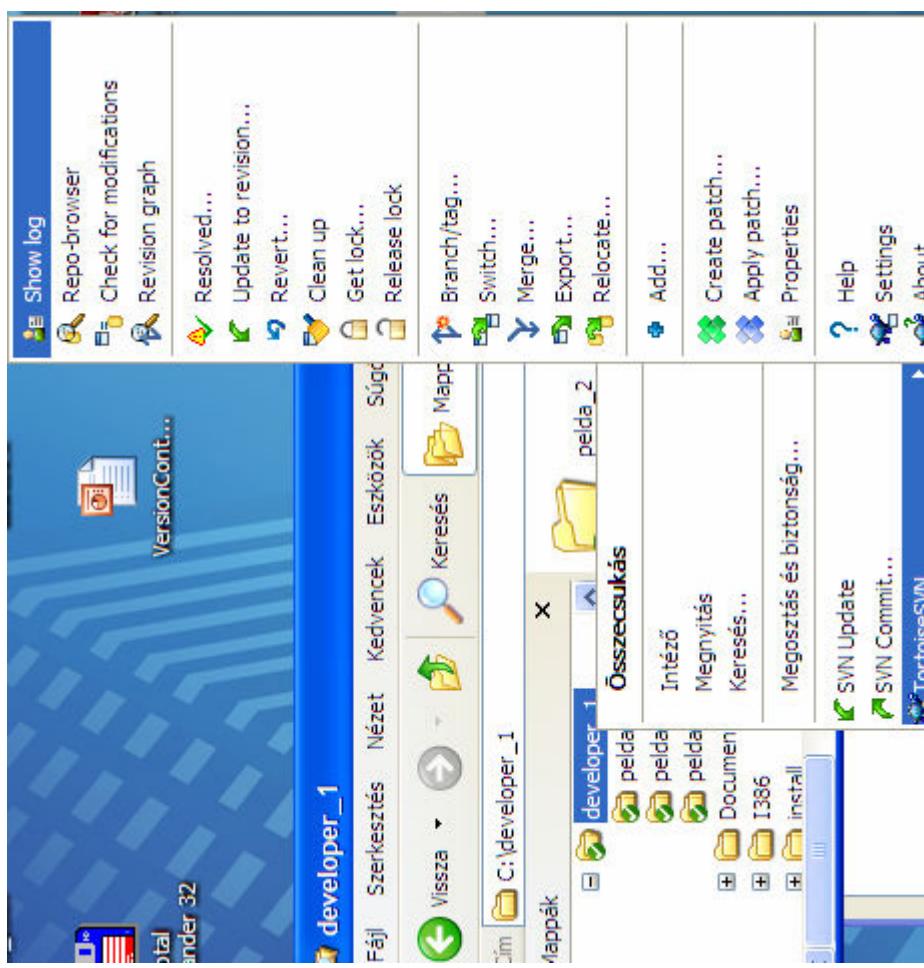
Theirs - pelda_3.c : Remote file

```
30 > .diag_printf("Thread.A.\r\n") ;>>>
31 > .diag_printf("Thread.B.\r\n") ;>>>
32 > .cyg_thread_delay(50) ;
33 >
34 }
35
36
37 /*
38 The.Thread.B
39 */
40 void.B_thread(cyg_addrword_t index)
41 {
42
43
44 while(1)
45 {
46   .diag_printf("Thread.B.asdad..\r\n");
47   .diag_printf("Thread.B.\r\n");
48   .cyg_thread_delay(100);
49 }
```

Több user módosít (Második lépés: Update, Merge, majd Commit)

- A módosított file-ok letöltése
- Update
- Ha nincs olyan változás, ami azt a részt érinti, amit mi használtunk, akkor automatikusan merge!
- Ha van, akkor konfliktust jelez, amit nekünk kell feloldani.

A verziók változtatásainak követése



A verziók változtatásainak követése

■ Mindegyikről meg lehet nézni

- Mi változott?
- Mikor?
- Ki által?
- Mi volt a komment?
- Össze lehet az egyes változásokat hasonlítni.

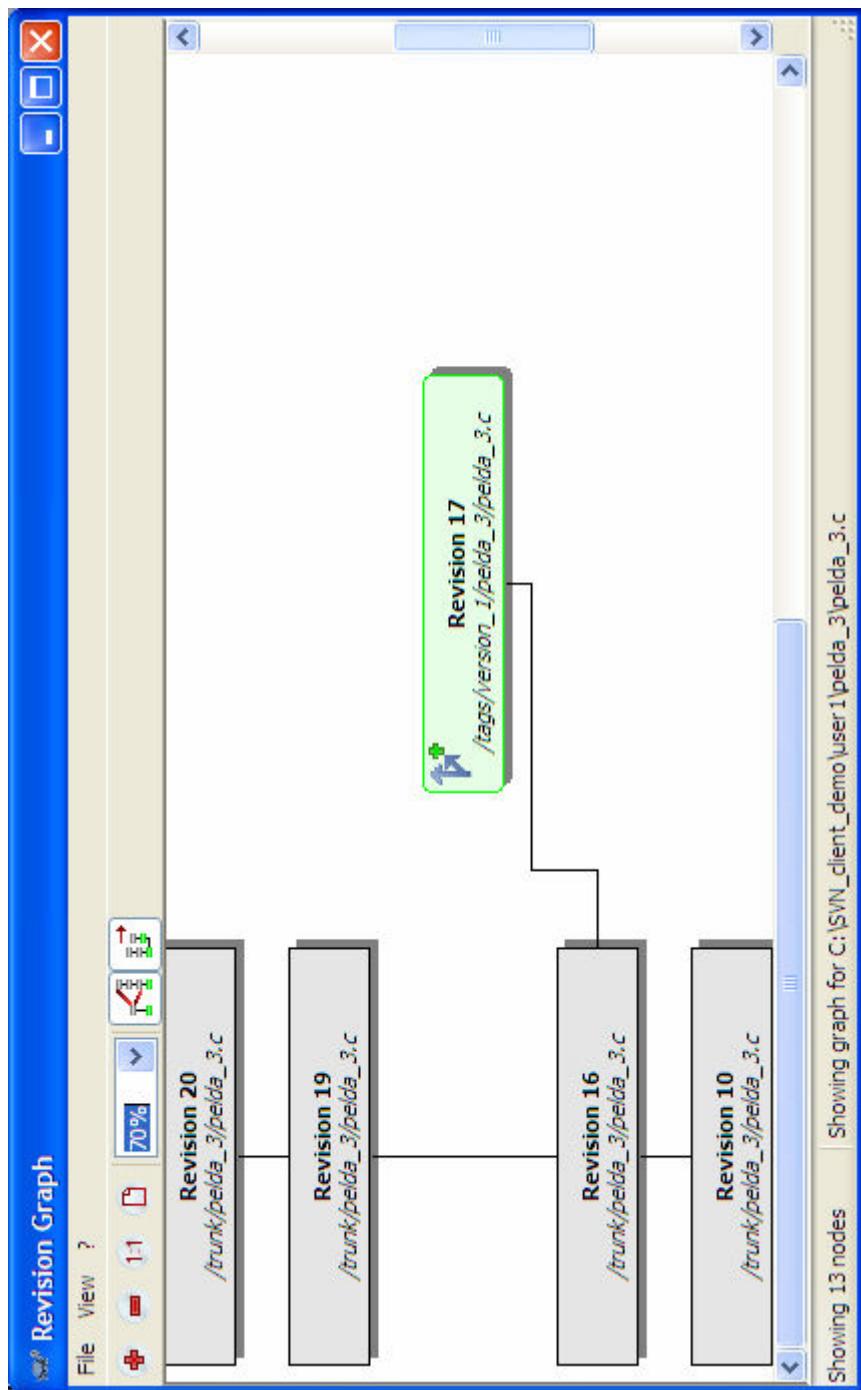
The screenshot shows a log viewer window titled "Log Messages - C:\developer_1". The interface includes a toolbar with icons for search, messages, authors, and paths. Below the toolbar are dropdown menus for "From" (set to 2007.11.27), "Actions" (set to 2007.11.29), and "To". The main area displays a table of log entries:

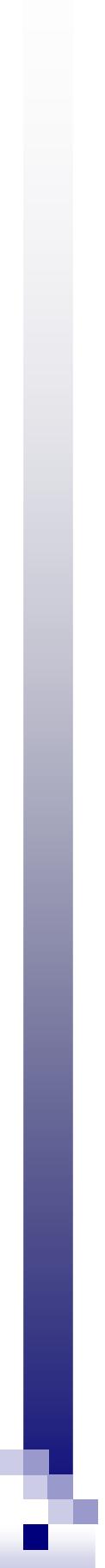
Revision	Actions	Author	Date	Message
14		user1	18:00:15, 2007. november 27.	deleted modification
13		user1	17:58:36, 2007. november 27.	parallel modified
12		user2	17:52:54, 2007. november 27.	second prints f modified too
11		user1	17:47:54, 2007. november 27.	first prints modified 3rd time
10		user2	17:45:44, 2007. november 27.	modified first time
9		user1	17:44:37, 2007. november 27.	printf modified second time
8		user2	17:22:11, 2007. november 27.	Greeting text has been modified
7		user1	17:15:23, 2007. november 27.	Move Remotely
6		user1	17:15:05, 2007. november 27.	Moved Remotely
5		user1	17:14:50, 2007. november 27.	Moved remotely
1		VisualSVN Server	14:29:19, 2007. november 27.	Initial structure.

Below the table, a note reads: "first prints modified 3rd time". To the right, there's a statistics section with buttons for "Statistics", "Help", and "OK". At the bottom, there are buttons for "Show All", "Next 100", and "Stop on copy/ rename".

A verziók változtatásainak követése grafikusan

■ Revision graph



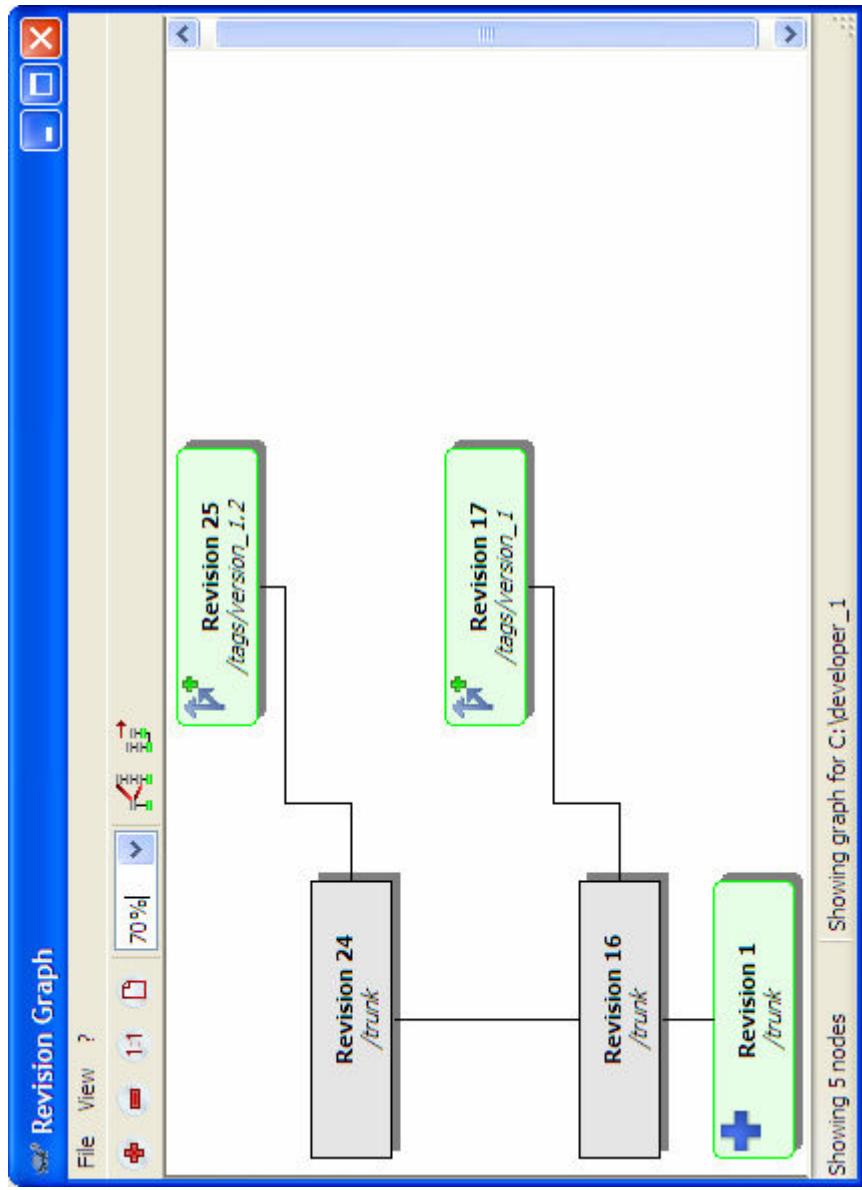


Visszatérés egy régi verzióhoz

- Update to revision

Új release létrehozása

- Branch/tag





Összefoglalás

- Hasznos eszköz, de
 - Nem tartja helyettünk számon, hogy mit, miért módosítottunk.
 - Csak segít a tiszta project adatbázis létrehozásában, nem csinálja meg helyettünk.
 - Nem helyettesíti a munkatársak közötti kapcsolattartást.
- Ezek az emberek dolgai.