Version Control System
-- base on Subversion 1.4

Sui Huang

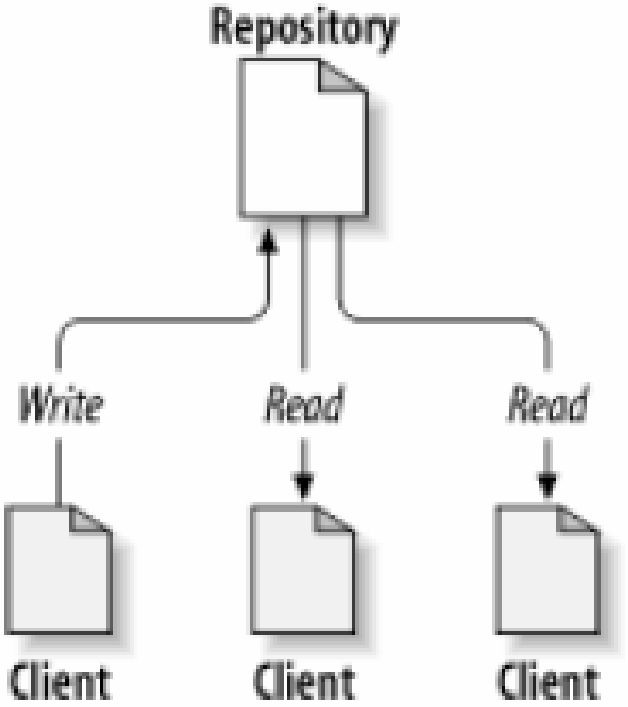
A tutorial for Software Engineering Course SE2AA4
Instructor: Dr. William M. Farmer
TAs: Clare So, Sui Huang, Jeffrey Heifetz

Jan 10th, 2006

Definition of Repository

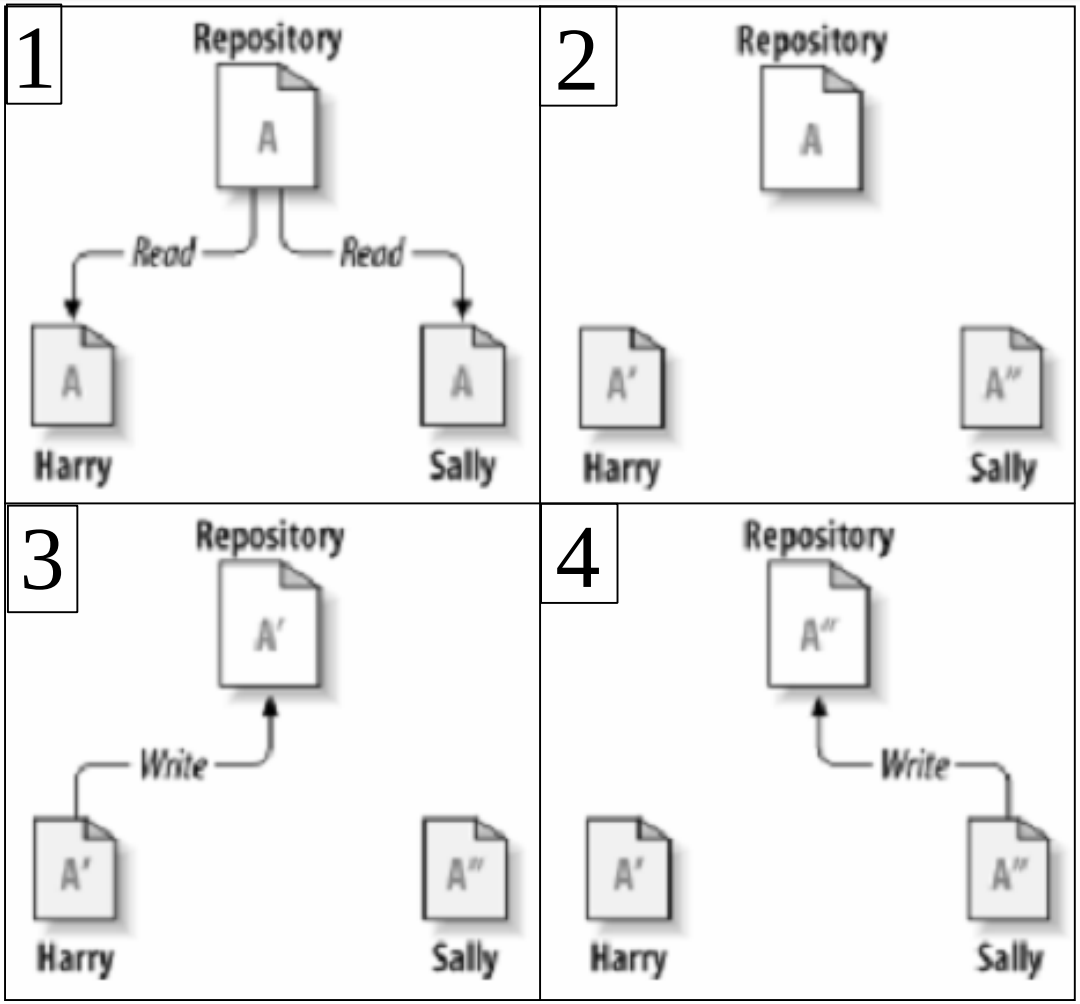
- A repository is a central store of data, for a centralized system of sharing information (e.g. Subversion)
- In large-scale or long-term software developments, a good repository can increase the effectiveness of communication, and save a lot of unnecessary documentation

Typical Client-server System



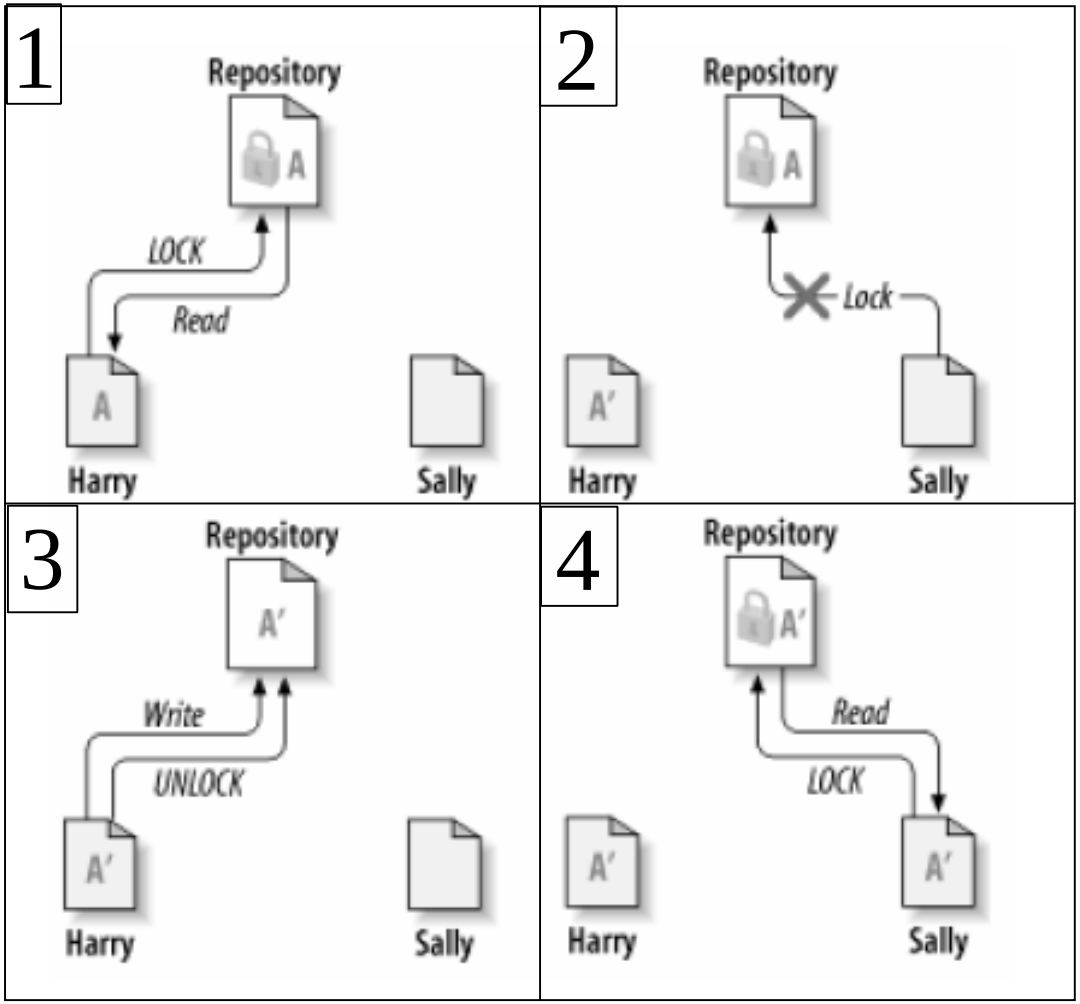
- Maybe good for typical file server or a simple web server
- Can cause problem for complicated collaborations in which multiple people writes to the repository

Typical Client-server System



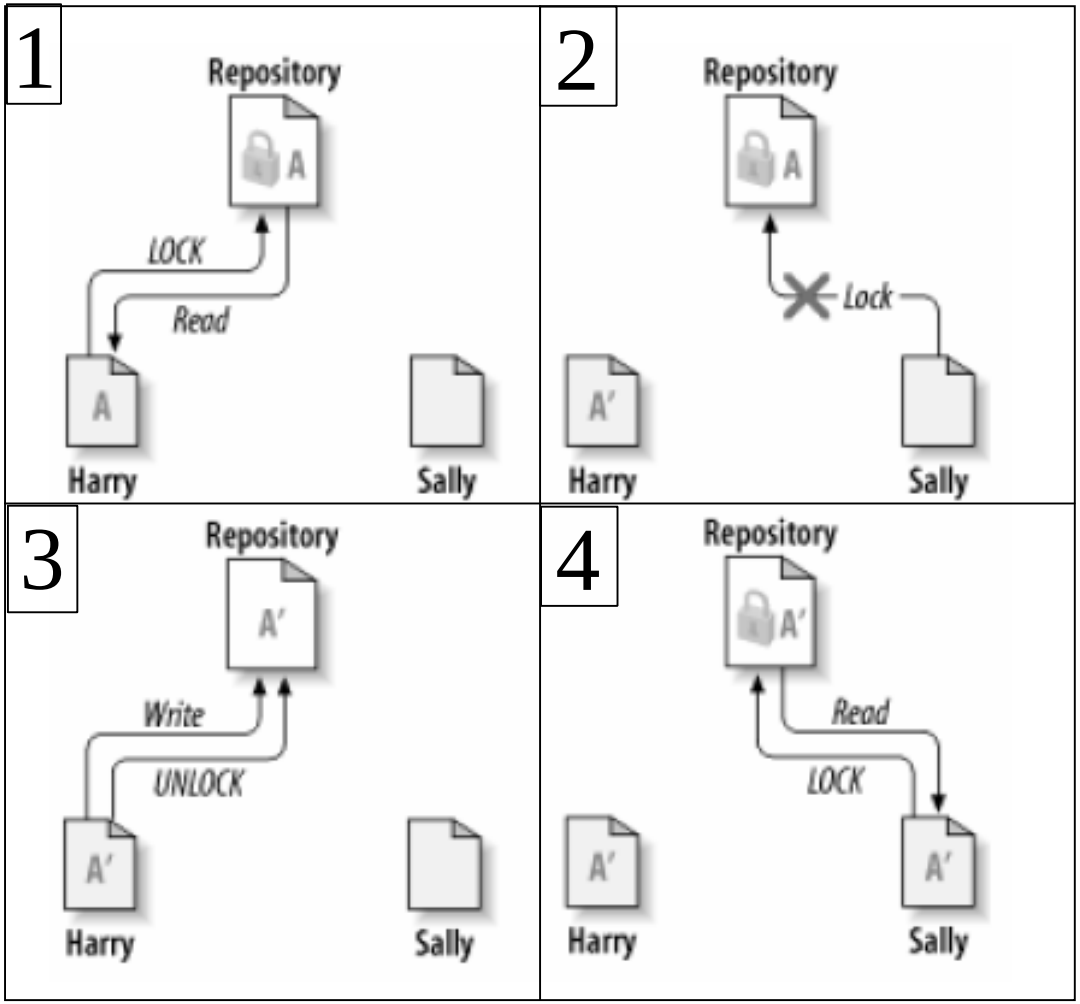
1. Two users read the same file
2. The both edit their copies
3. Harry publishes his version first
4. Sally accidentally overwrites Harry's version

Lock-Modify-Unlock Solution



1. Harry "locks" file A, then copies it for editing
2. While Harry edits, Sally's lock attempt fails
3. Harry writes his version, then release his lock
4. Now Sally can lock, read, and edit the latest version

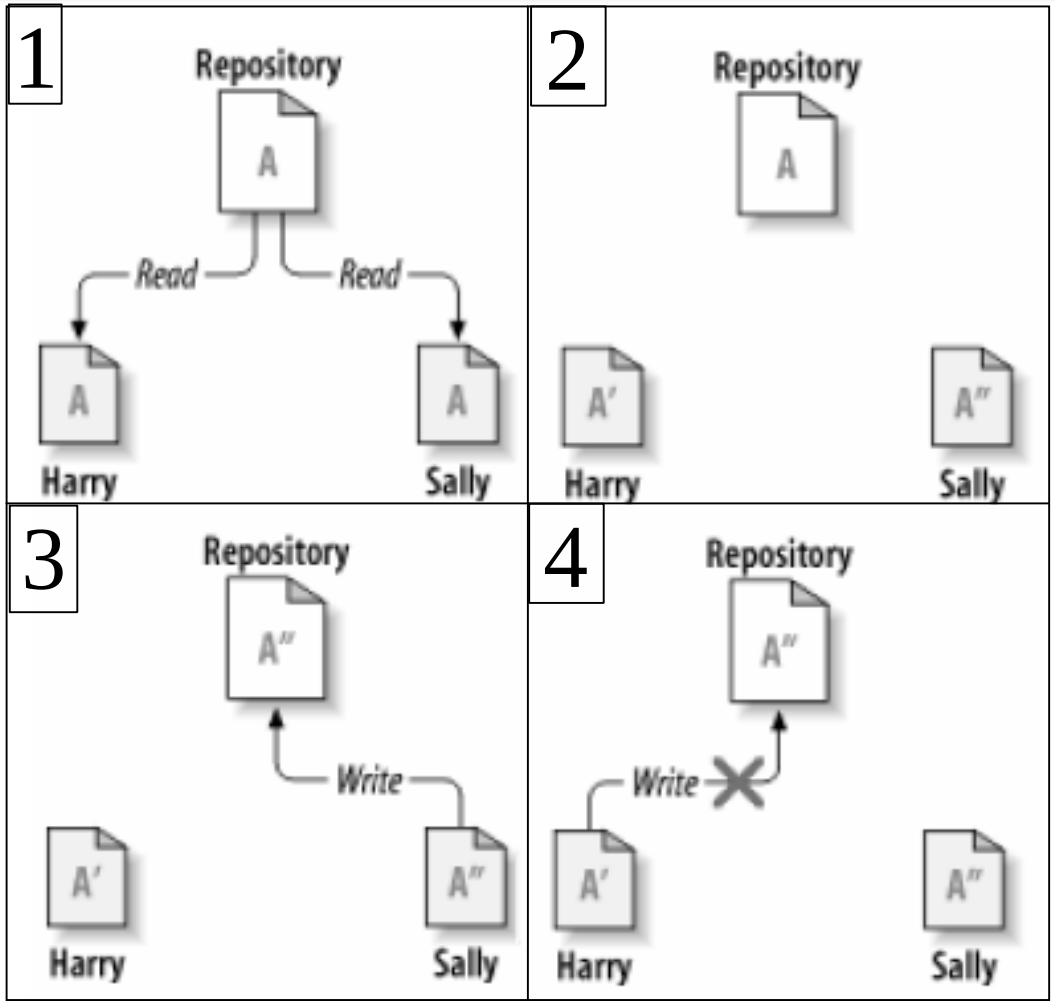
Lock-Modify-Unlock Solution



Drawbacks:

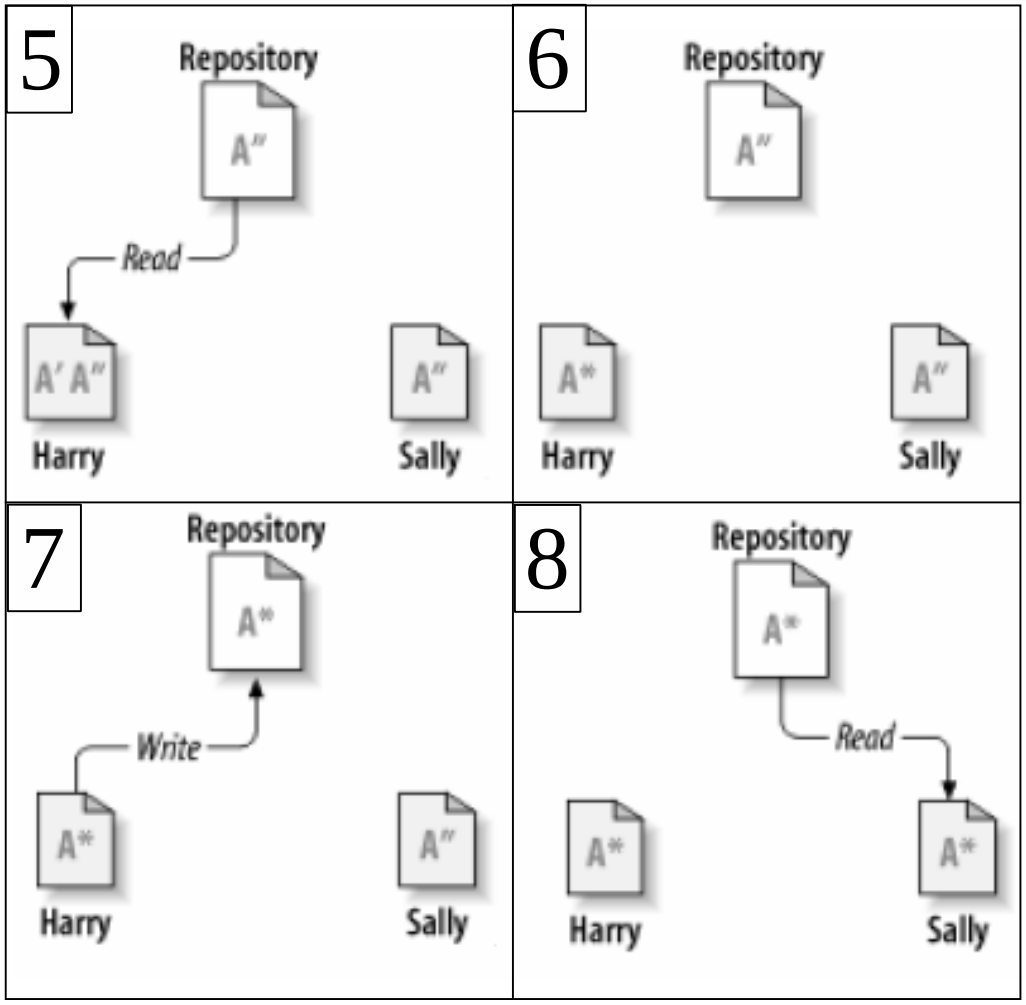
- Harry may lock and forget about it
- What about Harry want to edit the beginning of A but Sally want to edit the end?
- What if files A and B depends on one another, but Harry only lock A?

Copy-Modify-Merge Solution



1. Two users copy the same file
2. They both begin to edit their copies
3. Sally publishes her version first
4. Harry gets an “out-of-date” error

Copy-Modify-Merge Solution



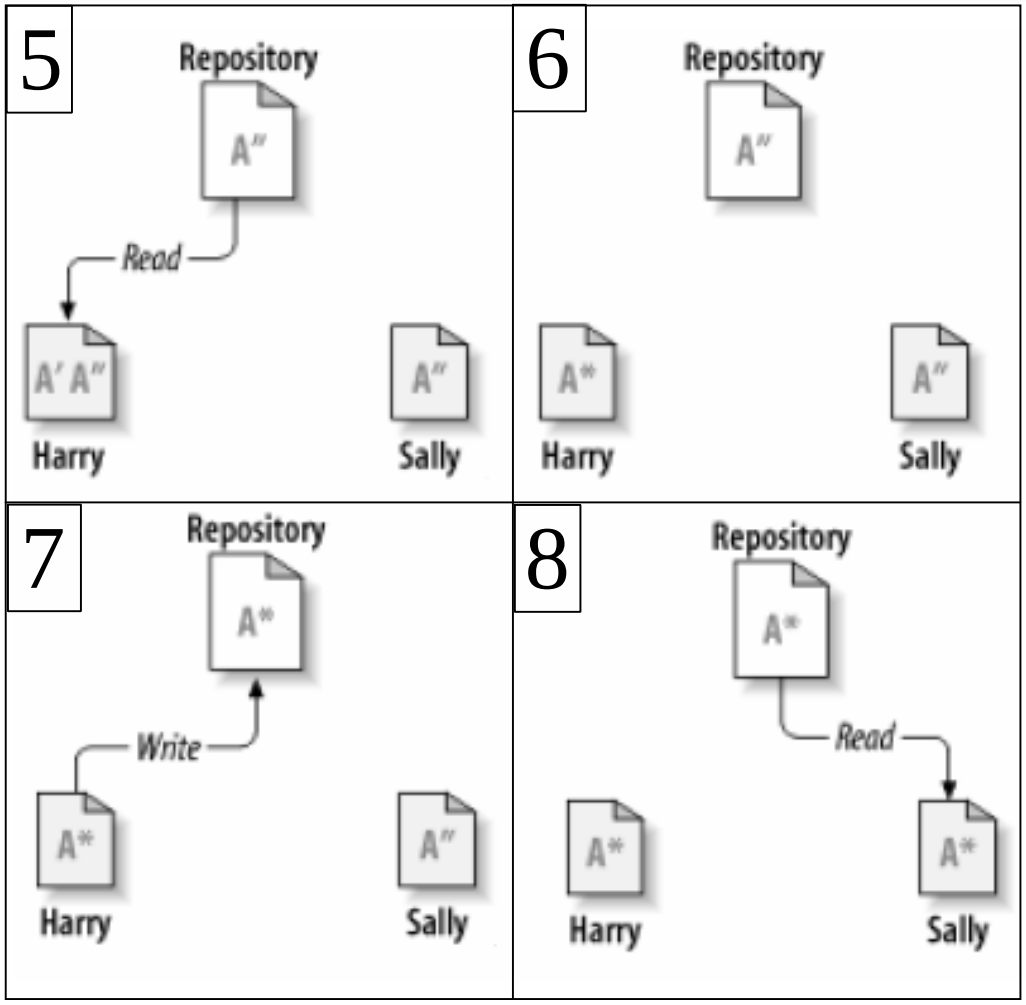
5. Harry compares the latest version to his own

6. A new merged version is created

7. The merged version is published

8. Now both users have each other's changes

Copy-Modify-Merge Solution



Note:

The situation is a conflict if Sally's changes overlap with Harry's changes.

To resolve the conflict Harry may need to discuss with Sally.

Software can merge files if there is not conflict between them.

Working Copy

The copies of files Harry and Sally download and work on with are working copies.

Core role of repository

The repository is the core of a Version Control System. Beside storing the most recent copy of files. It should be able to:

- Store all the previous revisions of the files
- Store all the branches of a files in parallel
- Log all who made the modifications, and when was the modifications (svn log command can check it)

Subversion can do all of these.

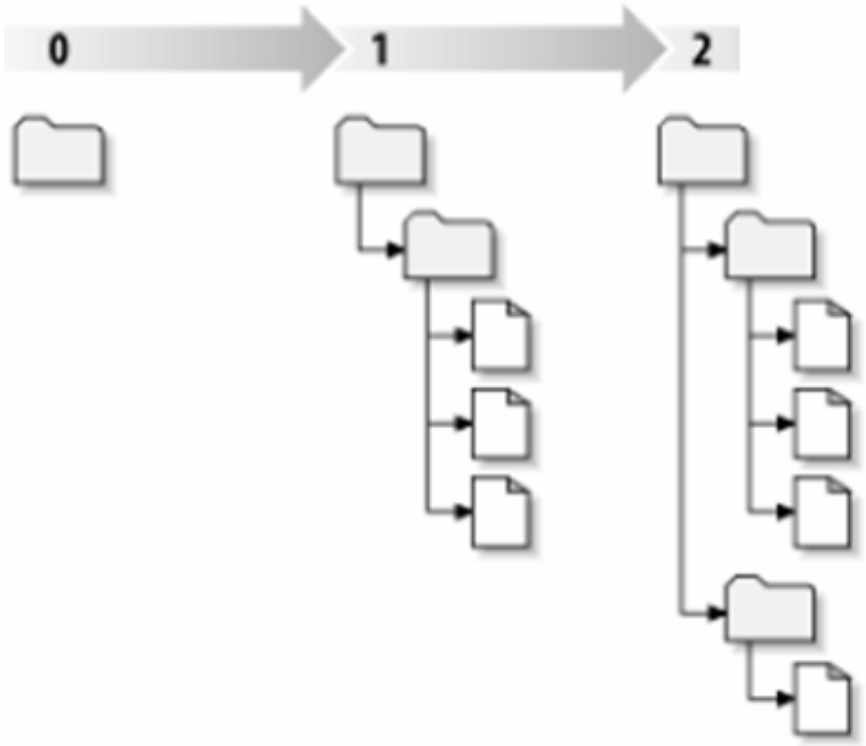
Revision

All the snapshots (called revisions) of the files are stored in the repository.

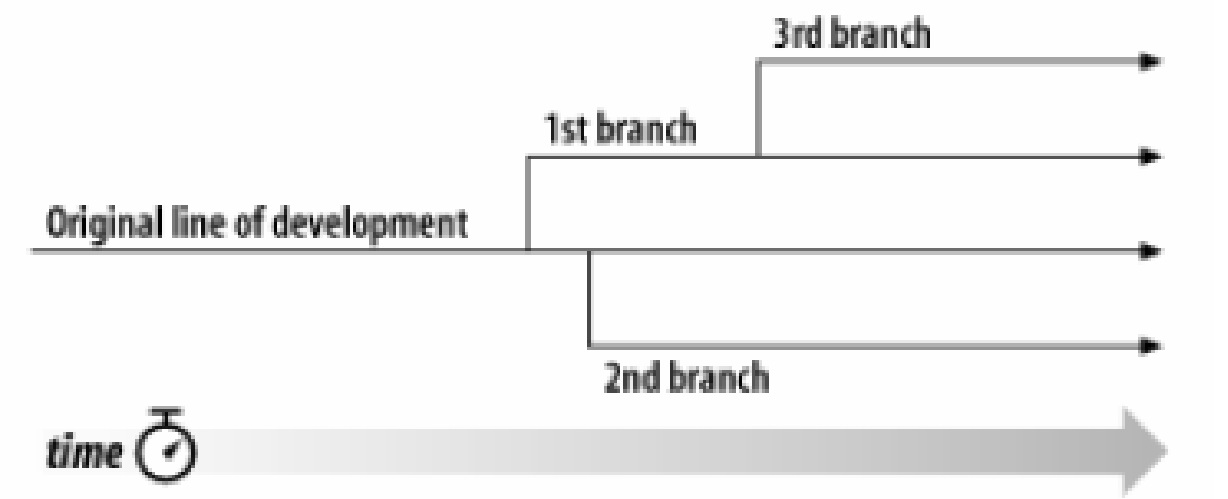
Integers are used as revision numbers in Subversion.

A client uses svn commit command to add a revision to the repository.

A client uses svn checkout with –-revision switch to obtain a previous revision.



Branch



Subversion allows you to create branches by copying your data; remember that the copies are related to one another, to help duplicate changes from one branch to another; there is more.

Repository URLs

URLs to identify repositories and sub-directories

```
$ svn checkout http://svn.example.com:9834/repos
```

Repositories can be local

```
$ svn checkout file:///path/to/repos
$ svn checkout file://localhost/path/to/repos
```

Both repositories and client can be on Windows System

```
C:\> svn checkout file:///X:/path/to/repos
C:\> svn checkout "file:///X|/path/to/repos"
```

Unsafe characters are not recommended

```
$ svn checkout "http://host/path with space/project/españa"
```

checkout / commit

Client obtain a working copies with svn checkout commands

```
$ svn checkout http://svn.example.com/repos/calc
A    calc/Makefile
A    calc/integer.c
A    calc/button.c
Checked out revision 56.

$ ls -A calc
Makefile  integer.c  button.c  .svn/
```

Client submit working copies with svn commit commands

```
$ svn commit button.c -m "Fixed a typo in button.c."
Sending        button.c
Transmitting file data .
Committed revision 57.
```

add / delete / copy / move

svn add foo

Schedule file or directory foo to be added to the repository. When you next commit, foo will become a child of its parent directory.

svn delete foo

Schedule file or directory to be deleted from the repository. If foo is a file, it is immediately deleted from your working copy. If foo is a directory, it will be removed from your working copy and the repository after commit.

svn copy foo bar

Create a new item bar as duplicate of foo.

svn move foo bar

Exactly the same as running “svn copy foo bar; svn delete foo”.

status / list

svn status command is for working copies

```
$ svn status --verbose
M                44        23    sally     README
                 44        30    sally     INSTALL
M                44        20    harry     bar.c
```

svn list command is for repositories

```
$ svn list http://svn.collab.net/repos/svn
README
branches/
clients/
tags/
trunk/
```

help

- There are more svn commands.
- Here is the most important commands:

  svn help

  List all the subcommands.

  svn help <subcommand>

  Describe syntax, switches, and behaviour of subcommand.

Basic Work Cycle

Get a up-to-date working copy: svn checkout or svn update

Make changes: svn add, svn delete, svn copy or svn move

Examine your changes: svn status, svn diff or svn revert

Merge others' changes into your working copy: svn update, svn resolved

Commit your changes: svn commit

Reference

Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael PilatoVersion, *Control with Subversion – For Subversion 1.4*, http://svnbook.red-bean.com/