

Version Control Systems



1. Periodically backed up
2. Central for all community
3. Accessible from anywhere

Version Control Systems



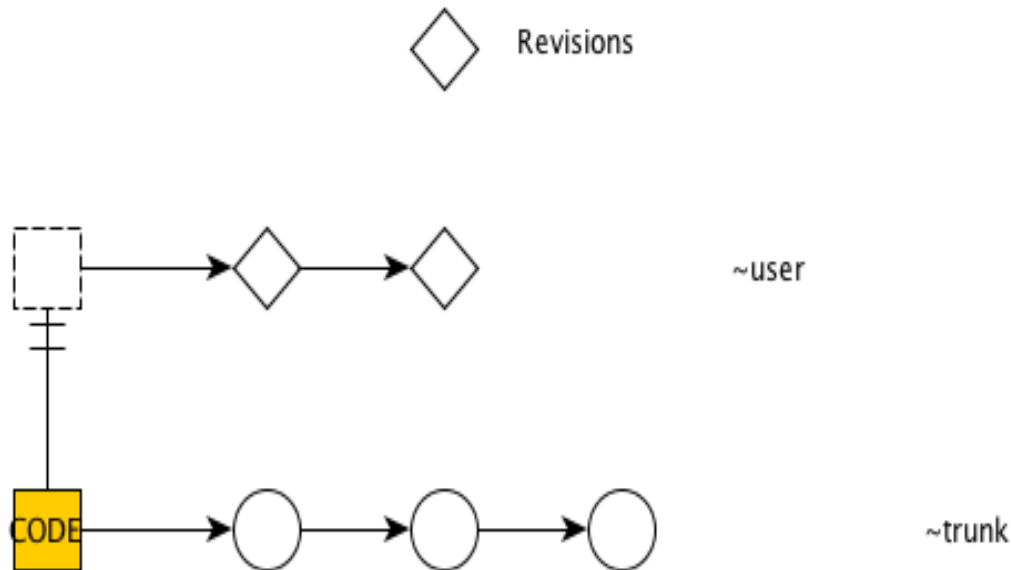
1. Periodically backed up
2. Central for all community
3. Accessible from anywhere

Version Control Systems



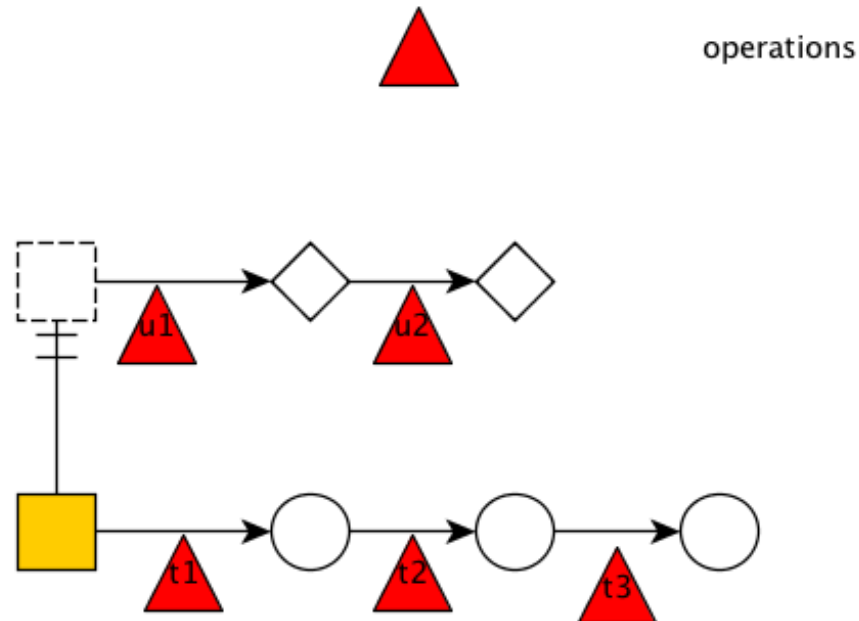
1. Pre-date the internet
2. **Manage group development**
 - automate as much as possible
 - readily share/access new developments

Simple svn repository



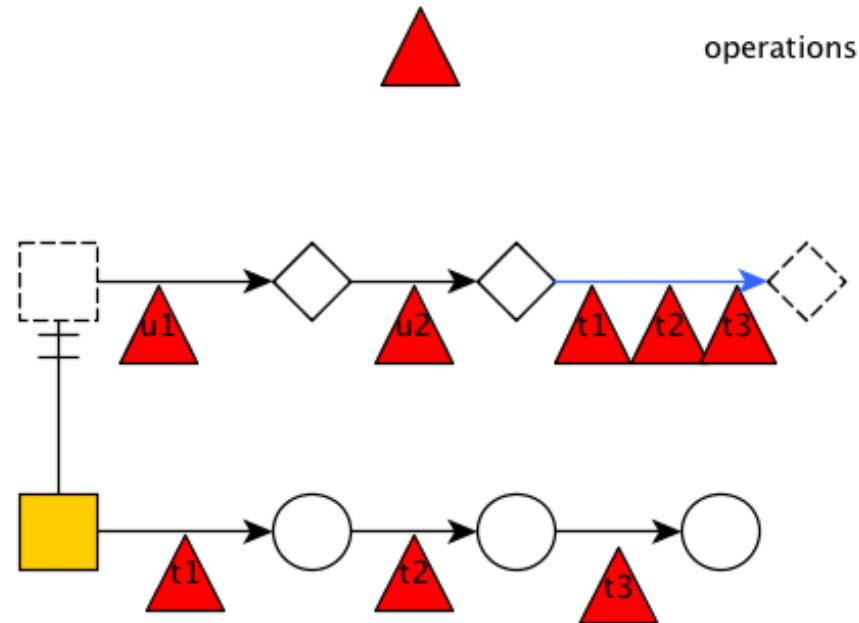
There is only one copy of the code
Revision nodes pictured are arbitrary

Simple svn repository



The useful part of svn records operations.
Consider user wants the trunk developments:

svn merge



From user: “*svn merge trunk*”

Performs the operations (t^*) recorded on trunk in the working copy (checkout of user, branched from trunk).

Typical Conflict

Example: on the same line in operations u1(mine) and t2(theirs)

<<<mine

! user initialize parameter

B = 0.0

=====

! trunk initializations

d=7.0

>>>theirs

BUT we want both

Conflicts are generally good

Example: realistic resolution

! user initialize parameter

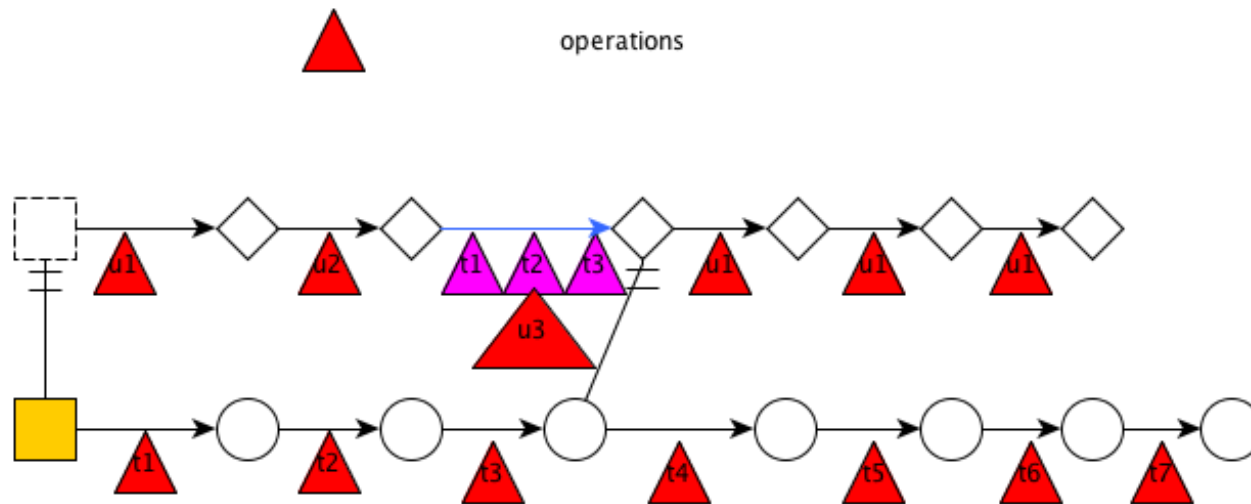
$B = 0.0$

! trunk initializations

$d=7.0$

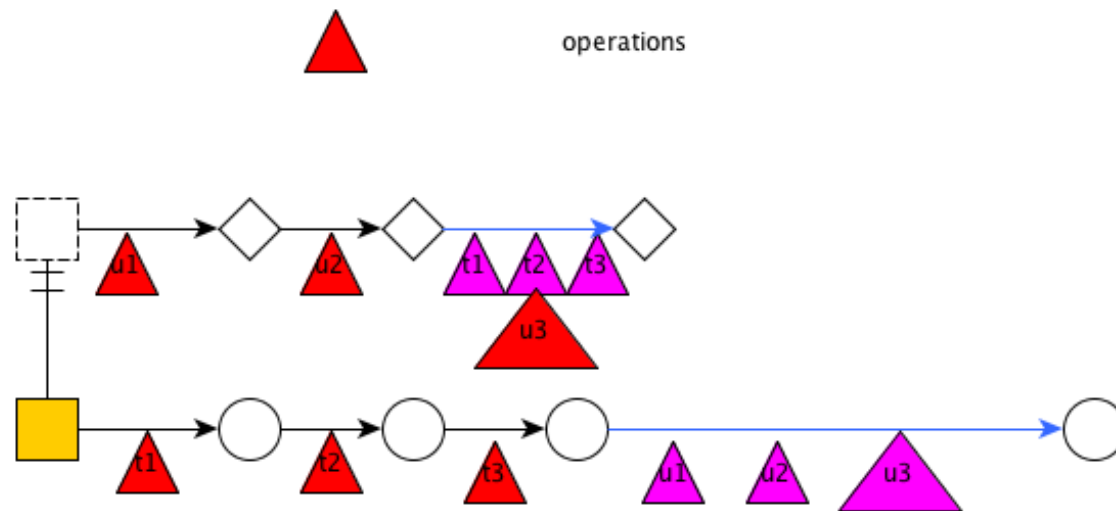
And commit

Moving on....



Performs the operations (t^*) since the last merger ONLY. This can continue forever until one day you decide to give back to the community....

svn merge



From trunk: “*svn merge user*”

Performs the operations (u^*) recorded on user in the working copy (checkout of trunk, branched from trunk).

BUT most of ours are bogus

There are genuine conflicts, but mostly we suffer bogus conflicts:

- manually entered changes
- copied changes

from the the trunk (or another user)

svn does recognize if an operation has no effect, however this is very limiting. For e.g. if an unrelated change has moved the line down by one.

svn merge

Most Conflicts could have been very simply avoided.
Moreover, conflicting t^* and u^* operations only occur –
SINCE branched or last merged

ONLY SINCE branched or last merged

- 1.svn commit regularly**
- 2.svn merge regularly**