

Version Control Software

Version control systems

- Version control (or revision control, or source control)
 - managing multiple versions of documents, programs, web sites, etc.
- CVS, Subversion, Mercurial, and Git
- Mercurial and Git - newer / better
- Concurrent Versions System (CVS)
- Apache Subversion (SVN)
- Git
- Mercurial

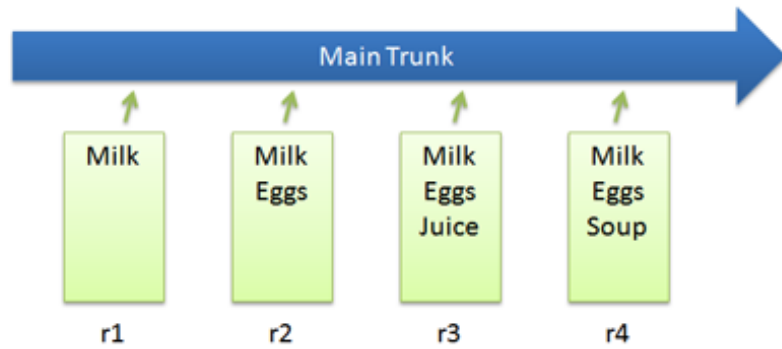
Why version control?

Originally for software developers – more and more for web developers including front-end web developers

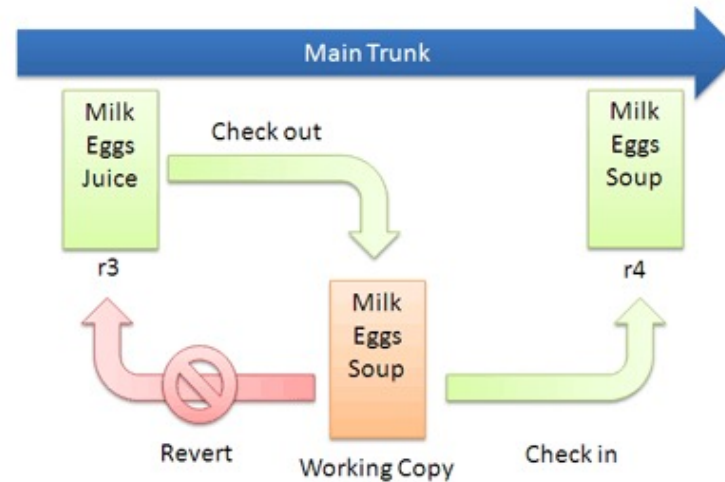
- For working by yourself
- For working with others
- For getting an internship or job

Check-ins, Checkouts, and Branching

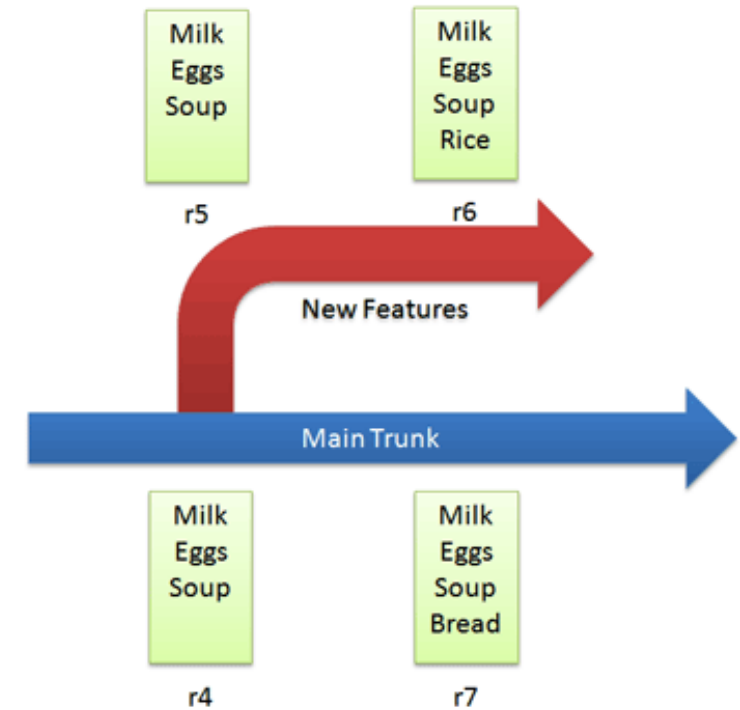
Basic Checkins



Checkout and Edit



Branching



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Why Git?

- Git has many advantages over earlier systems such as CVS and Subversion
 - More efficient, better workflow, etc.
 - See the literature for an extensive list of reasons
 - Of course, there are always those who disagree
- Best competitor: Mercurial
 - Same concepts, slightly simpler to use
 - The Eclipse plugin is easy to install and use
 - Much less popular than Git

Online (hosted) Git Solutions

- Github: github.com
- Bitbucket: bitbucket.org
- Heroku: heroku.com