Version Control System (VCS)

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

It allows you to revert files back to a previous state, revert the entire project back to a previous state, compare changes over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more.

Allows to

* revert files back to a previous state
* revert the entire project back to a previous state
* compare changes over time
* see who last modified something that might be causing a problem
* who introduced an issue and when, and more

Types:

* Local Version Control System
* Centralized Version Control System
* Distributed Version Control System
* Local Version Control System

Example – rcs

* Centralized Version Control System

Have a single server that contains all the versioned files, and a number of clients that check out files from that central place.

Example – cvs, subversion and perforce

CVS, Subversion, and Perforce, have a single server that contains all the versioned files, and a number of clients that check out files from that central place.

* + Cons
    - Centralized server – Crashes – Data is lost if there is no proper backup, all the data will be in local systems.
    - Server goes down for an hour then during that hour nobody can collaborate at all or save versioned changes to anything they’re working on.
* Distributed Version Control Systems

Check out the latest snapshot of the files: they fully mirror the repository

Any of the client repositories can be copied back up to the server to restore it.

Every checkout is really a full backup of all the data