9. tétel Version control

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. If you are a graphic or web designer and want to keep every version of an image or layout (which you would most certainly want to) version Control allows you to revert selected files back to a previous state. Using a VCS also generally means that if you screw things up or lose files, you can easily recover.

For almost all software projects, the source code must be protected. Version control protects source code from both catastrophe and the casual degradation of human error and unintended consequences.

Software developers working in teams are continually writing new source code and changing existing source code. The code for a project, app or software component is typically organized in a folder structure or "file tree". One developer on the team may be working on a new feature while another developer fixes an unrelated bug by changing code, each developer may make their changes in several parts of the file tree.

Version control helps teams solve these kinds of problems, tracking every individual change by each contributor and helping prevent concurrent work from conflicting. Changes made in one part of the software can be incompatible with those made by another developer working at the same time. This problem should be discovered and solved in an orderly manner without blocking the work of the rest of the team. Further, in all software development, any change can introduce new bugs on its own and new software can't be trusted until it's tested. So testing and development proceed together until a new version is ready.

Good version control software supports a developer's preferred workflow without imposing one particular way of working. Ideally it also works on any platform, rather than dictate what operating system or tool chain developers must use. Great version control systems facilitate a smooth and continuous flow of changes to the code rather than the frustrating and clumsy mechanism of file locking - giving the green light to one developer at the expense of blocking the progress of others.

Benefits

One of the most popular VCS tools in use today is called Git. Git is a *Distributed* VCS, a category known as DVCS.

1. A complete long-term change history of every file. This means every change made by many individuals over the years. Changes include the creation and deletion of files as well as edits to their contents.
2. Branching and merging. Team members work at the same time on different parts of the code, creating a "branch" and when they are finished they merge that work back together, enabling developers to verify that the changes on each branch do not conflict.
3. With VCS you can trace each change made to the software and connect it to project management and bug tracking software such as Jira, and being able to annotate each change with a message describing the purpose and intent of the change.