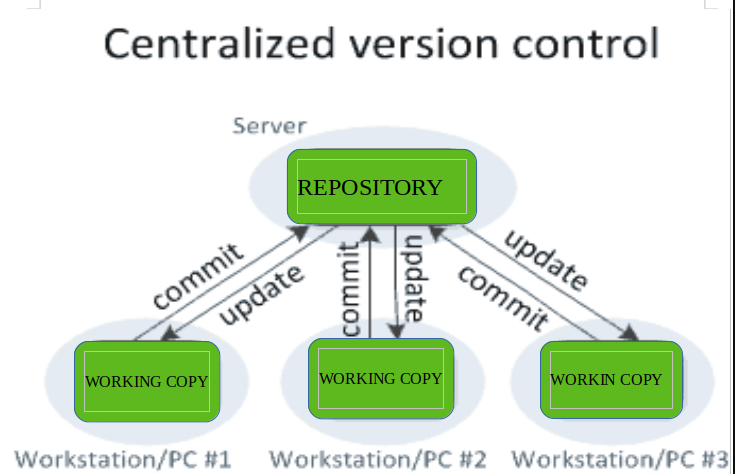
A **source control system**, also known as a **version control system** (VCS), is a software tool that tracks and manages changes to source code and other text files. It's a vital part of the software development process, helping teams work more efficiently and smarter.



Here are some benefits of using a source control system:

* **Track changes**: A VCS keeps a history of code changes, so you can revert to previous versions if needed.
* **Resolve conflicts**: A VCS can help resolve conflicts when merging contributions from multiple sources.
* **Collaborate**: A VCS allows developers to collaborate on code and isolate their work until it's ready.
* **Troubleshoot**: A VCS can help identify who made changes and what the changes were.
* **Protect source code**: A VCS protects source code from unintended consequences and human error.

Some popular VCS options include: Git, Mercurial, SVN, and preforce

**These are common Git commands used in various situations:**

start a working area (see also: git help tutorial)

**clone** Clone a repository into a new directory

**init** Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

**add** Add file contents to the index

**mv** Move or rename a file, a directory, or a symlink

**restore** Restore working tree files

**rm**  Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

**bisect**  Use binary search to find the commit that introduced a bug

**diff** Show changes between commits, commit and working tree, etc

**grep** Print lines matching a pattern

**log** Show commit logs

**show** Show various types of objects

**status** Show the working tree status

grow, mark and tweak your common history

**branch**  List, create, or delete branches

**commit** Record changes to the repository

**merge** Join two or more development histories together

**rebase** Reapply commits on top of another base tip

**reset**  Reset current HEAD to the specified state

**switch**  Switch branches

**tag** Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

**fetch** Download objects and refs from another repository

**pull** Fetch from and integrate with another repository or a local branch

**push** Update remote refs along with associated objects

**'git help -a**' and **'git help -g**' list available subcommands and some

concept guides. See **'git help <command>**' or **'git help <concept>**'

to read about a specific subcommand or concept.

See **'git help git**' for an overview of the system.