# **What is Version Control?**

* The purpose of **version control** is to allow software teams track changes to the code, while enhancing communication and collaboration between team members.
* Version control facilitates a continuous, simple way to develop software.
* **Source code** acts as a single source of truth and a collection of a product’s knowledge, history, and solutions.
* Version control (or code revision control) serves as a safety net to protect the source code from irreparable harm, giving the development team the freedom to experiment without fear of causing damage or creating code conflicts.
* If developers code concurrently and create incompatible changes, version control identifies the problem areas so that team members can quickly revert changes to a previous version, compare changes, or identify who committed the problem code through the revision history.
* With version control systems, a software team can solve an issue before progressing further into a project. Through **code reviews**, software teams can analyze earlier versions to understand how a solution evolved.
* Depending on a team's specific needs, a version control system can be local, centralized, or distributed. A local version control system stores files within a local system. Centralized version control stores changes in a single server. A distributed version control system involves cloning a Git repository.

# **What Is a Version Control System?**

* A **version control system** (VCS) is a system that tracks changes to a file or set of files over time.
* Developers can check out a file from the server, make changes, and check the file back in. The server then stores the new version of the file.

# **Distributed Version Control System**

* A **distributed version control system** (DVCS) is a type of version control system that allows users to access a repository from multiple locations.
* DVCSs are often used by developers who need to work on projects from multiple computers or who need to collaborate with other developers remotely.

# **What is Git?**

* Git is an open-source distributed version control system that is that lets you track changes you make to your files over time.
* With Git, you can revert to various states of your files (like a time traveling machine).
* You can also make a copy of your file, make changes to that copy, and then merge these changes to the original copy.
* For example, you could be working on a website's landing page and discover that you do not like the navigation bar.
  + But at the same time, you might not want to start altering its components because it might get worse.
  + With Git, you can create an identical copy of that file and play around with the navigation bar.
  + Then, when you are satisfied with your changes, you can merge the copy to the original file.
* You are not limited to using Git just for source code files – you can also use it to keep track of text files or even images.
* This means that Git is not just for developers – anyone can find it helpful.

# **What is the difference between Git vs. GitHub?**

* Git is a version control system that lets you manage and keep track of your source code history.
* GitHub is a cloud-based hosting service that lets you manage Git repositories.

Diagram, timeline

Description automatically generated

**How to use Git?**

**General Introduction**

<https://alistapart.com/article/get-started-with-git/>

<https://education.github.com/git-cheat-sheet-education.pdf>

**How To Remove Untracked Files from the Working Tree**

<https://stackoverflow.com/questions/61212/how-to-remove-local-untracked-files-from-the-current-git-working-tree>

**Add DS\_Store To .gitignore**

<https://gist.github.com/lohenyumnam/2b127b9c3d1435dc12a33613c44e6308>