Git is a distributed version control system (VCS) that is commonly used for tracking changes in source code during software development. It was created by Linus Torvalds in 2005 and has since become one of the most widely used version control systems in the world. Here are some key features and concepts associated with Git:

1. Version Control: Git allows developers to track changes made to their code over time. It keeps a historical record of all changes, making it easy to review and manage the development process.

2. Distributed: Git is a distributed VCS, which means that every developer has a complete copy of the repository, including its entire history. This decentralization allows for greater collaboration and flexibility in development workflows.

3. Repositories: A Git repository is a collection of files and a database that stores information about those files, including their complete history. There can be a local repository on your machine and remote repositories hosted on services like GitHub, GitLab, or Bitbucket.

4. Commits: Commits are the fundamental building blocks of a Git repository. A commit represents a snapshot of the project at a specific point in time. Each commit has a unique identifier and contains changes made to the code.

5. Branches: Git allows you to create branches to work on new features or changes independently. Branches are lightweight and can be easily merged back into the main codebase when the work is complete.

6. Merging: Merging is the process of integrating changes from one branch into another. Git provides various merging strategies, such as fast-forward, recursive, and three-way merge.

7. Pull Requests: In the context of platforms like GitHub or GitLab, a pull request (or merge request) is a feature that allows developers to propose changes and request that they be reviewed and merged into the main codebase.

8. Conflict Resolution: When multiple developers make changes to the same part of the code, Git can detect conflicts during merging. Developers need to resolve these conflicts manually.

9. GitHub, GitLab, Bitbucket: These are web-based platforms that provide hosting for Git repositories, along with collaboration and project management features. They are commonly used for open-source and private software development.

10. Command Line and GUI Tools: Git can be used through the command line with commands like `git init`, `git clone`, `git add`, `git commit`, and more. There are also graphical user interfaces (GUIs) available for those who prefer a visual way of interacting with Git.

Above are some of the functions used via Git.

GitHub is a web-based platform for software development and version control using Git. It's a widely popular and highly influential service that has significantly impacted the way developers collaborate on projects.

Some if its properties include:

1. Version Control with Git:

- GitHub is built on top of Git, a distributed version control system. Git allows developers to track changes in their code, maintain a history of their work, and collaborate effectively.

2. Repository Hosting:

- GitHub provides a platform for hosting Git repositories. Developers can create repositories to store their code and collaborate with others.
- It offers both public and private repository options, making it suitable for open-source projects and proprietary code.

3. Collaboration and Workflow:

- GitHub facilitates collaboration among developers. Multiple contributors can work on a project simultaneously.
- It offers tools like pull requests, issues, and project boards for managing tasks and discussing code changes.
- Collaborators can review code, suggest improvements, and discuss issues, all within the GitHub platform.

4. Code Review:

- One of GitHub's strengths is its robust code review system. Developers can create pull requests to propose changes, and others can review, comment on, and approve these changes before merging.
- This ensures high code quality, helps catch bugs early, and maintains project integrity.

5. Social Features:

- GitHub introduces social networking features to coding. Developers can "star" repositories, follow other developers, and discover popular projects.
- It encourages community engagement and networking among developers.

6. Issue Tracking:

- GitHub's issue tracking system allows developers to report and manage bugs, feature requests, and other tasks.
- Developers can assign issues, label them, and link them to code changes, providing a comprehensive project management solution.

7. Continuous Integration and Deployment (CI/CD):

- GitHub integrates with various CI/CD tools and services, allowing automated testing and deployment pipelines for software projects.
- It helps maintain code quality and streamline the release process.

8. Community and Open Source:

- GitHub is a hub for open-source development. Many popular open-source projects are hosted on GitHub, making it easy for anyone to contribute and collaborate.

- It has become a central platform for open-source software development, offering a place for communities to grow and thrive.

9. APIs and Integrations:

- GitHub offers a rich set of APIs, enabling developers to integrate GitHub functionality into their own applications and workflows.
- It integrates with various development tools and services, making it a versatile part of the development ecosystem.

10. Education and Learning:

- GitHub provides educational resources, tutorials, and the GitHub Education program to help students and teachers learn and teach software development collaboratively.