Git is a distributed version control system that is widely used for tracking changes in source code during software development. There are various tools and packages that complement Git or provide additional features. Here are some popular ones:

1. **GitKraken:**
   * GitKraken is a cross-platform Git GUI client that provides an intuitive interface for interacting with Git repositories.
   * Website: [GitKraken](https://www.gitkraken.com/)
2. **SourceTree:**
   * SourceTree is a free Git GUI client for Windows and macOS that simplifies Git operations with a visual interface.
   * Website: [SourceTree](https://www.sourcetreeapp.com/)
3. **GitHub Desktop:**
   * GitHub Desktop is an official Git GUI client provided by GitHub. It's user-friendly and integrates with GitHub features.
   * Website: [GitHub Desktop](https://desktop.github.com/)
4. **GitLab:**
   * GitLab is a web-based Git repository manager that provides source code management (SCM), continuous integration, and more.
   * Website: [GitLab](https://about.gitlab.com/)
5. **Bitbucket:**
   * Bitbucket is a Git repository management solution by Atlassian. It includes features like code collaboration, CI/CD, and more.
   * Website: [Bitbucket](https://bitbucket.org/)
6. **Git Bash:**
   * Git Bash is a command-line interface for Git on Windows. It provides a Unix-like shell experience and Git command-line capabilities.
   * Website: [Git for Windows](https://gitforwindows.org/)
7. **Magit:**
   * Magit is a Git interface for Emacs, providing a powerful set of features for managing Git repositories within the Emacs editor.
   * Website: [Magit](https://magit.vc/)
8. **Git LFS (Large File Storage):**
   * Git LFS is an extension for managing large files in Git repositories. It replaces large files with text pointers in the Git history.
   * Website: [Git LFS](https://git-lfs.github.com/)
9. **Hub:**
   * Hub is a command-line wrapper for Git that makes it easier to work with GitHub repositories directly from the command line.
   * Website: [Hub](https://hub.github.com/)
10. **GitFlow:**
    * GitFlow is a branching model for Git that defines a strict branching structure designed to facilitate collaboration and release management.
    * Website: [GitFlow](https://nvie.com/posts/a-successful-git-branching-model/)
11. **GitLab Runner:**
    * GitLab Runner is the open-source project that is used to run your jobs and send the results back to GitLab. It is used in GitLab CI/CD pipelines.
    * Website: [GitLab Runner](https://docs.gitlab.com/runner/)
12. **Git Cheat Sheet:**
    * Not a tool, but a handy reference. Git Cheat Sheets provide quick reference guides for common Git commands.
    * Example: [GitHub Git Cheat Sheet](https://github.github.com/training-kit/downloads/github-git-cheat-sheet/)

Remember that the choice of Git tools depends on your preferences, workflow, and the specific requirements of your projects. Git itself is a powerful tool, and these additional tools can enhance your experience and productivity with version control.