# **Bellevue College expands on Git functionality with Git Going**

### **Bellevue College’s code reviewer is a web-based interface which makes it easier for programmers to upload and to collaboratively review source codes with the help of the Git version control system.**

*October 13th, 2020*, Bellevue College is now developing a software designed to assist programmers in tracking changes in a particular project with Git’s version control and comparison features. For example, a software development team is working on one project such as building a website; each team member has a different workflow, and this makes it difficult to collaborate while being productive. However, the best way to get the project done is by creating a repository where every member will be able upload and update code files all at the same time without having to slacken the task implementation.

Git’s version control makes this possible with its branching and merging capabilities. The branching feature enables programmers to duplicate the source code and to safely change it without altering the original repository. The merging feature is important once the developer gets their code running properly; before the code is merged to the main codebase and made official, it needs to be reviewed and approved by other team members. Note that the author and any changes committed are being tracked, and changes can be reverted when needed. These capabilities bring benefits to the workflow of one programmer working on his own project locally, up to a hundred programmers working on maintaining an already distributed service.

Many beginning programmers are intimidated by Git and its commands. Our goal is to create an environment which is intuitive and efficient in helping users to keep track of their code changes in both individual and collaborative contexts, benefiting from all the advantages that Git has to offer, such as the ability to work offline or revert to previous versions, all while using your own workflow.

Set your fears aside with Git Going, as we simplify the process of code review and provide you the tools you need to control, and master, your code! Just enter “Site Name” into the browser of your choice and get reviewing!

“*Giving users easier, more intuitive version control service is the only thing I have ever considered to be worthy of my time.*” Said Travis Lamb, Co-Co-Co-Co-Senior member of Team GonnaGetOurDegrees. “*It’s kept me up at night for years. Pondering the problems, and solutions, the ways to make code review more accessible. Really, working on this project has been a dream come true since Day 1.*”

“*I can’t believe how little I knew about Git, and GitHub when I started using “Git Going*””, said every student in every CS class ever. “*In the short time since, “Git Going” has redefined what version control systems mean to me!*”

# **FAQ**

1. **How do I use Git Going?**

**Proceed to “Site Name” and register a new account! Authenticate your e-mail address and start working on your project!**

1. **Do I have to pay to use Git Going?**

**Absolutely not! “Git Going” has been, and always will be, a free service!**

1. **Are more features being considered for Git Going?**

**Absolutely! New features are constantly being added, and old features expanded, to continue to provide you with the smoothest code comparison and review process possible!**

1. **Do I need a GitHub account?**

**No! Just register an account in the Git Going site and start right away! Possible future integration of GitHub accounts with Git Going is being considered!**

1. **Is Git Going open source?**

**Git Going is a non-profit open source work!**

**Internal FAQ**

1. **Why do users need Git Going?**

**Code review can be a complicated process that overwhelms programmers, especially first-time users. Git Going seeks to simplify the process, pruning back the façade of difficulty to reveal the essentially simple process beneath. Users will especially like the simplified approach offered to them, as user experience is a primary focus of Git Going.**

1. **Why do users not need Git Going?**

**Unfortunately, it is true to say that the market of code review software and sites can be perceived as bloated, and not in need of more developers, rather refinement of existing structures. Users may prefer other code review tools for features not implemented in Git Going.**

1. **How scalable is Git Going?**

**Overall, there are many possible features that may be eventually added to the system. Integration with GitHub, advanced document review for many different formats, possible automated code review, student/teacher modes for educational purposes, pipelining between GitHub or other Code Review software allowing for easy merge and fork management. Realistically, many of these features may not be achieved, but are theoretically possible.**

1. **How extendable is Git Going?**

**Requirements change frequently. It could be users who require some more features for better efficiency of their work. Thus, the necessity of taking future growth into consideration by making code resistant to change, which means one will be able to add new functionalities without modifying the already-existing ones. The project is essentially using the open-closed principle; open for extension, closed for modification, assuring that units of code do not have to be impaired. Furthermore, the code is built on high cohesion and loose coupling principles making it easier to maintain.**

1. **What does Git Going offer over its competitors?**

**What “Git Going” brings to the table is its own diverse range of tools for simplifying the code review process, as well as an intuitive design that practically coaches its users through the review, approval and merging processes; many code reviewers today have clunky, hard to follow UIs which lose customers before they even get to see the features of the product! Equally, as a free product, it has a leg up on competition that charges for their tool.**

**As “Git Going” is designed to be utilized in educational structures, role-based access control features could be used allowing educators and students to only access the information they need.**

**Another benefit that can’t be overstated is our web development approach. Many code reviewers which don’t utilize a cloud solution for storage require an on-premise implementation, and even many of those which do offer cloud solutions require specific platforms. By utilizing a cloud solution for storage, and making our interface a web app, we can truly say we are cross-platform.**

1. **What limitations may Git Going face in development?**

**While “Git Going” can keep version history of all files, not all files may support text reviewing (e.g. an Excel document). More particularly, Git Going may only accept and review documents that are also acceptable by GitHub. This includes a majority of the most used text files and documents. Another important limitation to consider is that while Git Going may be capable of much more than simple code review, it may face developmental constraints on how many other features can be added.**