Tierney Coren

Professor Martin

English 111

28 October 2012

Benefits of Git for Version Control

Git is a free and open-source standard that is small and fast at what it does, created by Linus Trovalds (the creator of Linux.)[1] (McMillan, 11 May 2012) Git has brought the notion of a local repository that one updates as one codes; whereas, the older methods, such as Subversion, cvs, vault, and so on, use the model of uploading code to a main source code server when one thinks it is ready for production.[2] (Geurts, 05 May, 2008) In the last few years, Git has become the most ubiquitous method to control versions and source spanning the entire development industry. The main method of perpetuation of the Git system has been GitHub, whose system has brought this standardization of version and source control to the developer ecosystem.

To name one competitor, Subversion, which was a former forerunner in the field of version control, is at a disadvantage (as many other forms of version control management are) when compared to Git in terms of useability, as said by Michael Stum in a StackOverflow question on the topic. There are many situations, such as a power outage, air travel, and being disconnected from the main repository, where a developer is without an Internet connection or a connection to the version control server and still needs to write code and add it to the repository to maintain a level of organization and clarity within said code. Subversion is implemented so that one must have a connection with the server to update one's repository; Git, on the other hand, allows someone to have an instance of their repository on every machine they (and anyone else that has the permission to view it) need to access it from. When the developers are able to connect to the Internet or the contact the main repository again, they can use a single command to update it to what they have done while offline.

Despite its ability to be decentralized, Git can still be used as a centralized code hosting and maintenance tool. GitHub is most likely the best example of a centralized Git-based service (although others do exist, such as BitBucket[4] [Atlassian, 2012]) that we can use today. There is a slight problem with using Git as a centralized service—one still has to go through the motions of the decentralized process of saving to one's local repository and then updating the centralized repository with the newest code from one's local one. This is one trade off one has to decide on when first considering using Git—should one **c**hoose to use the immense functionality of Git and accept going through small, seemingly needless steps to update one's code, or should one choose a centralized source control management system that allows one to update in one fell swoop while losing a good amount of functionality. GitHub has 2.5 million[5] (GitHub, 2012) developers hosting 8.1 million code repositories,[6] (GitHub, 2012) as shown in statistics on their Features pages, which are both extraordinary numbers and a testament to how many developers think that Git is an extraordinary version control management system, even when used in a centralized manner.

There's another benefit of Git that helps many projects grow: its capability to fork.[3] (Stum, 28 Dec. 2009) Forking is the process of pulling code from a master repository into one's own personal repository. When one has made whatever changes one wanted to, one can make a pull request, which is a request for one's newly added code to be pulled back into the master repository, to have the updates uploaded. This provides enormous power to one particular type of project: open-source software. Open-source software is empowered by pull requests because they allows anybody to take the code base for the project, add to it, and send it back to the people who maintain the main project repository for use in production software. This is by far the most common method of usage on GitHub simply because GitHub's pricing plans are set up so that anyone can create public (oftentimes forked) repositories for free', but one must pay to have access to private repositories. Quite a few huge projects that are very influential are hosted in public repositories on GitHub: jQuery, Rails, Redis, Phone Gap, Node, phpBB, Ruby, Django, Linux, and many more, as seen on GitHub's Interesting Repositories page.[7] (GitHub, 2012) There are, of course, also uses for forking outside of the open-source world. It is useful in essentially the same manner in an enterprise environment, as it allows multiple people or teams to work on the same project together, with extreme ease. This is made clear by GitHub's offering of a private, self-hosted version of its software, called GitHub:Enterprise. This software is an exact replica of the GitHub website, except it has been built for Enterprise-level companies to host on their own servers and use privately.

The benefits of using Git are clear. The ability that Git yields to keep one's code structured and updated even when one don't have access to one's main repository is invaluable to those that are without power, that travel, or that don't constantly have access to the main repository. The strength that Git provides to simultaneously be used in a centralized and a decentralized fashion is universally useful for any type of developer. The amazing features of Git for code iteration, namely via forking, are helpful to open-source projects as well as Enterprise-level users to separate development of code, yet unify the results. All of these factors make Git a clear forerunner in today's field of version control systems.

Works Cited

[4] BitBucket.org. Atlassian. 2012. Web. 28 Oct. 2012. <https://bitbucket.org/>

[5] Features/Community・ GitHub, 2012. Web. 28 Oct. 2012. <https://github.com/features/community>

[6] “Features/Hosting” GitHub, 2012. Web. 28 Oct. 2012. <https://github.com/features/hosting>

[2] Geurts, Jim. “The benefits of using Git for source control.” Blog post. Web. 05 May, 2008. 28 Oct. 2012. <http://biasecurities.com/2008/05/the-benefits-of-using-git-for-source-control/>

[7] Interesting Repositories・ GitHub, 2012. Web. 16 Nov. 2012. <https://github.com/repositories>

[1] McMillan, Robert. “Linus Torvalds Invented Git, But He Pulls No Patches With GitHub”・*Wired.* Wired.com, 11 May 2012. Web. 28 Oct. 2012. <http://www.wired.com/wiredenterprise/2012/05/torvalds_github/>

[3] Stum, Michael. “Why is Git better than Subversion?” Online posting. 28 Dec. 2009. *StackOverflow*. Web. 28 Oct. 2012. <http://stackoverflow.com/questions/871/why-is-git-better-than-subversion#answer-875 >