

Collaborating in GitHub

Nicky Bleiel

IBM Watson

nableiel@us.ibm.com

Abstract - GitHub (<https://github.com/>) is a web-based repository for software projects and is reportedly the world's largest open source community, hosting over 31 million repositories that include both code and the documentation for that code. GitHub includes development tools such as issue tracking (bug reports and feature requests), notifications, diffs, and status dashboards; as well as social features. There are three options for documenting a project in GitHub: Readmes, a wiki, and GitHub Pages. All of these options can be authored in Markdown syntax. Because docs are stored in the same repository as code, they can follow the same workflow, which is especially useful in an Agile environment.

Index Terms – Collaboration, documentation, GitHub, markdown

IT ALL STARTS WITH GIT

GitHub uses Git as a repository. Git (initially released in 2005) is a distributed version control system for software development that was developed by Linus Torvalds, creator of Linux. According to a survey released in May 2014 by the Eclipse Foundation, Git is the #1 code management tool, with one third of software developers using it [1]. When using Git, documentation can have the same workflow as code, because docs can be stored in the same repository, and reviewed along with the code. This visibility is positive in general, but especially useful in an Agile software development environment, where docs are part of the “definition of done.”

WHY GITHUB?

While Git is a command-line tool, GitHub has a user interface, plus includes a number of other useful features.

But those aren't the primary reasons for GitHub's popularity. The main reason is because it has simplified the process of contributing to open source projects. According to Gregg Pollack of Code School (in the article “What Exactly Is GitHub Anyway?” [2]), before GitHub, contributing to a project meant a developer had to download the source code, make changes, create a patch,

then email that patch for consideration. With GitHub, you can create a “fork” of the source code, make your changes, and submit a “pull” request for consideration. And there is a public record of all of this, so you can build a reputation in the community.

GitHub could be considered a “one-stop-development-shop” because it includes version control, issue tracking (bugs and feature requests), documentation, notifications, diffs, and status dashboards. It also has social features. You can “follow” other contributors and GitHub will automatically alert you of their activities. You can also “watch” specific projects — you will receive notices about those also. If you just want to tag a repository, but skip the notifications, you can “star” it. You can then go to your “stars page” to catch up on those projects, as well as take a look at the “stars” of your friends.

Organizations can post projects on the public version of GitHub for free, as well as purchase GitHub Enterprise for their internal use behind a firewall. Individuals can create public repositories and contribute to projects with a free account, but must pay a fee to create private repositories that are accessible only to them and their invited collaborators.

All of these factors, plus the need to work collaboratively with team members around the world, combine to explain the popularity of GitHub.

GITHUB TERMINOLOGY

Selected GitHub terminology (from the GitHub Glossary) [3]:

- **Repository** (also referred to as a “repo”): A repository is the most basic element of GitHub. They're easiest to imagine as a project's folder. A repository contains all of the project files (including documentation), and stores each file's revision history.
- **Fork**: A fork is a personal copy of another user's repository that lives on your account. Forks allow you to freely make changes to a project without affecting the original.
- **Pull request**: Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators.

- **Merge:** Merging takes the changes from one branch (in the same repository or from a fork), and applies them into another.

GITHUB DOCUMENTATION OPTIONS

GitHub provides several ways to document a project: Readmes, GitHub Pages, and a Wiki.

I. Readmes

When you create a project repository in GitHub, a Readme file is created automatically. It is edited in Markdown, and has the file extension of .md. Markdown is a lightweight markup language that can be converted to HTML easily. If you have written wiki syntax, it will look very familiar. Markdown was originally developed in 2004 by John Gruber, and has splintered into different variations. GitHub uses “GitHub Flavored Markdown” which has extra features like syntax highlighting, task lists, tables, and @mentions (a quick way to notify someone that you need them to take a look at something). GitHub provides a quick guide to mastering Markdown that includes a downloadable quick reference guide. Markdown is also used for GitHub Wikis and Pages, for commenting, and when creating sharable GitHub reusable snippets, called Gists.

Because the readme for a GitHub repository is included with the code, members of the community may suggest changes to the readme that you need to review and merge. You can also review the readmes in other public repositories and suggest changes to them.

II. Wikis

You can expand upon the information in the Readme by adding wiki pages to your project repository. Wiki pages are easy to create, just click the Wiki button in your repository. You can edit in Markdown, or choose one of the other eight edit modes. Wiki pages are stored in Git repositories like all other content. By default, anyone can edit your wiki, but you can change the settings and make your wiki read-only. If your wiki is public, you should monitor contributions from the community.

III. GitHub pages

Another way to create documentation for your project repository is to use GitHub Pages. These are webpages hosted and published on GitHub. They are authored in Markdown, and you can use GitHub-provided themes to create a custom look. You can even add your Google Analytics tracking ID to each of your Pages. After you are satisfied with your content and the theme, you can publish. By default, the URL of your GitHub pages will be: [http://\[accountName\].github.io/\[repoName\]](http://[accountName].github.io/[repoName]). GitHub Pages are always public, even if your repository is private. You can also create separate GitHub Pages for you or your organization.

MANAGING ISSUES IN GITHUB

You can manage documentation issues (bug and feature requests) in a project using GitHub’s issue tracking. If you have used other issue tracking systems, many of the features will be familiar, and as with any system, you need to think about how granular issues should be, how you want to tag them, and what best practices your team should follow. GitHub issues can be authored in Markdown, so you can add formatting, create task lists to track progress, use emojis, and more.

To view and manage your issues on a virtual task board (which is great for agile shops and virtual teams), check out ZenHub, a productivity tool that can be integrated with GitHub.

USING GITHUB FOR VERSION CONTROL

GitHub has, at its core, a version control system. Files are stored in Git repositories and if you are familiar with using Git for version control, that knowledge will be useful when working in GitHub. A quick reference to Git commands can be found here: <https://git-scm.com/docs>.

Storing documentation files along with code and other project artifacts means that docs are not siloed in GitHub, and follow the same workflow as code. In GitHub, the project’s collaborators can review and contribute to documentation files.

INTEGRATIONS WITH OTHER TOOLS AND SUPPORT FOR STUDENTS

Over 70 productivity tools can be integrated with GitHub, including Slack, ZenHub, and Travis CI. See the list at <https://github.com/integrations>. GitBook can be used to host and write documentation, see <https://github.com/integrations/gitbook>. There is a GitHub Student Developer Pack that provides more than 15 free development tools for students at <https://education.github.com/pack>.

BUT THERE’S MORE

GitHub’s origins are in software collaboration, but it can be used to collaborate on other types of projects. It is now being used to collaborate on projects as diverse as Gregorian chants, licensing agreements, and wedding invitations [4]. For additional resources, see [5-11].

REFERENCES

- [1] Ian Skeritt. (2014). *Eclipse Community Survey 2014 Results* [Online]. Available: <https://ians.kerrett.wordpress.com/2014/06/23/eclipse-community-survey-2014-results/>
- [2] Klint Finley. (2012, July 14). *What Exactly Is GitHub Anyway?* (TechCrunch) [Online]. Available:

<http://techcrunch.com/2012/07/14/what-exactly-is-github-anyway/>

[3] GitHub Glossary. (GitHub Help) [Online]. Available: <https://help.github.com/articles/github-glossary/>

[4] R. McMillan. (2013, September 2). *From Collaborative Coding to Wedding Invitations: GitHub Is Going Mainstream* (Wired) [Online]. Available: <http://www.wired.com/2013/09/github-for-anything/>

[5] P. Bell and B. Beer. *Introducing GitHub*. Sebastopol, CA: O'Reilly Media, 2014.

[6] S. Chacon and B. Straub. *Pro Git. (2nd Edition)* New York, NY: Apress, 2014.

[7] GitHub Guides. *Mastering Markdown* [Online]. Available: <https://guides.github.com/features/mastering-markdown/>

[8] GitHub Training and Guides. *The Basics of Git and GitHub* [Online]. Available: <https://www.youtube.com/watch?v=U8GBXvdmHT4>

[9] GitHub training (try the free 15 minute “Basics of Git” exercise) [Online]. Available: <https://training.github.com/>

[10] GitHub Octodex. (Original Octocat plus variations) [Online]. Available: <https://octodex.github.com/>

[11] A. Andersson. *Emoji Cheat Sheet* [Online]. Available: <http://www.emoji-cheat-sheet.com/>

ABOUT THE AUTHOR



Nicky Bleiel is a Watson Information Developer at IBM and a GitHub user. She is a Past President and Associate Fellow of the Society for Technical Communication and has over 20 years of experience writing and designing content for software products in a variety of industries. She is a popular speaker at many conferences, including the STC Summit, WritersUA, tcworld, CIDM, and LavaCon; and has been published in STC's Intercom, tcworld magazine, ISTC Communicator, and more. Learn more about her at nickybleiel.com.