

What Method Do You Use to Propose Code Changes? Using GitHub to Collaborate

Prof. Monika Rokade¹ Miss. Maya R. Shinde² Prof. Sunil Khatal³

^{1,3}Assistant Professor ²ME Students

^{1,2,3}Department of Computer Engineering

^{1,2,3}Sharadchandra Pawar College of Engineering, Otur, Pune, India

Abstract— GitHub (<https://github.com/>) is a web-based software project repository that claims to be the world's largest open source community, with over 31 million repositories containing both code and documentation. GitHub contains social elements as well as development tools like as issue tracking (bug reports and feature requests), notifications, diffs, and status dashboards. Readmes, a wiki, and GitHub Pages are the three alternatives for documenting a project in GitHub. Markdown syntax may be used to create all of these choices. Because documents and code are both saved in the same repository, they may share the same workflow, which is extremely valuable in an Agile setting.

Keywords: Collaboration, Documentation, GitHub, Markdown

I. INTRODUCTION

While Git is a command-line programme, GitHub provides a user interface and a variety of other capabilities.

However, these aren't the main reasons for GitHub's success. The key reason is that it has made contributing to open source projects more easier. Before GitHub, a developer had to download the source code, make changes, build a patch, and then email that patch for review, according to Gregg Pollack of Code School (in the article "What Exactly Is GitHub Anyway?" [2]). You can create a "fork" of the source code on GitHub, make your modifications, and then submit a "pull" request for review. All of this is documented in the public record, allowing you to establish a reputation in the community.

Because it contains version control, issue tracking (bugs and feature requests), documentation, notifications, diffs, and status dashboards, GitHub may be regarded a "one-stop-development-shop." It contains social features as well. You can "follow" other contributors, and GitHub will notify you of their activity automatically. You may also "follow" specific projects and receive notifications about them. You can "star" a repository if you only want to tag it and not receive notifications. You may then go to your "stars page" to catch up on those projects as well as check out your friends' "stars."

Organizations can upload projects for free on GitHub's public edition, or pay GitHub Enterprise for corporate use behind a firewall. Individuals with a free account can build public repositories and contribute to projects, but they must pay a price to create private repositories that are only accessible by them and their invited collaborators. All of these features, combined with the necessity to collaborate with team members all across the world, explain why GitHub is so popular.

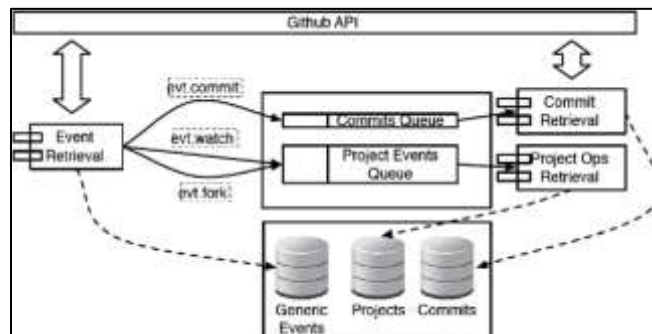


Fig. 1: System Architecture

II. BACKGROUND AND TERMINOLOGY

A. Repository:

The most basic component of GitHub is a repository. It's easier to think of them as a project's folder. A repository holds all of the project files (including documentation) as well as the revision history for each file. You can own repositories on your own or share ownership with others in your business. The visibility of a repository can be adjusted to limit who has access to it. See "About repository visibility" for further details.

You can grant collaborator access to other people in user-owned repositories so that they can work on your project. If a repository is controlled by an organisation, you can grant access permissions to members of that organisation to collaborate on it. See "Permission levels for a user account repository" and "Repository permission levels for an organisation" for further details.

You can work with an unlimited number of collaborators on an unlimited number of public repositories with a full feature set, or an unlimited number of private repositories with a limited feature set, with GitHub Free for user accounts and organisations. Upgrade to GitHub Pro, GitHub Team, or GitHub Enterprise Cloud to gain advanced tooling for private repositories. See "GitHub's Products" for further details.

B. Fork:

A fork is a copy of another user's repository that you keep on your account. You can make changes to a project without affecting the original via forks.

A GitHub fork is a clone of a repository (repo) that is stored in your account instead of the account from which the data was forked. You own your forked copy after you fork a repo. This implies you can make changes to your forked repository without affecting the parent repository's content.

C. Pull Request:

Pull requests are user-submitted modifications to a repository that are either approved or denied by collaborators.

Pull requests allow you to notify others about changes you've made to a branch in a GitHub repository. You can debate and review potential modifications with collaborators and make follow-up commits before your changes are merged into the base branch once a pull request is filed.

source repository. Docker images that use such a service have a link to their GitHub/Bitbucket source on Docker Hub. On their summary page, they provide extra image information, such as the Dockerfile of the most recently successfully produced image and the build status.

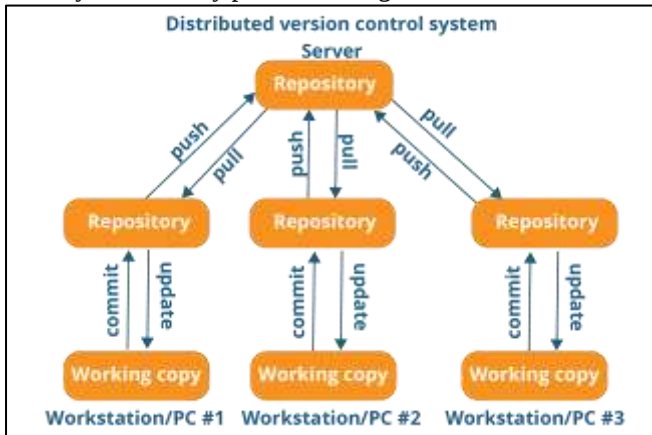


Fig. 2:Distributed Version Control System Git,Github

D. Merge:

Merging is the process of applying changes from one branch (in the same repository or from a fork) to another.

Git's approach of bringing a forked history back together is called merging. The git merge command allows you to combine the several development lines established by git branch into a single branch.

III. GITHUB DOCUMENTATION OPTIONS

Readmes, GitHub Pages, and a Wiki are all options for documenting a project on GitHub.

A. Readmes:

A Readme file is automatically created when you establish a project repository on GitHub. It's written in Markdown and has a .md file extension. Markdown is a simple markup language that can be easily transformed into HTML. It will look extremely similar to anyone who has written wiki syntax. Markdown was created by John Gruber in 2004 and has now branched into numerous variants.

GitHub employs "GitHub Flavored Markdown," which includes syntax highlighting, task lists, tables, and @mentions, among other things (a quick way to notify someone that you need them to take a look at something). A brief tutorial to mastering Markdown is available on GitHub, along with a downloadable quick reference guide. Markdown is also used for GitHub Wikis and Pages, commenting, and making GitHub Gists (sharable reusable snippets).

Members of the community may suggest changes to the readme, which you must examine and merge, because the readme for a GitHub repository is included with the code. You may also look over the readmes in other public repositories and make suggestions for improvements.

B. Wikis:

By adding wiki pages to your project repository, you can build on the information in the README. Simply click the Wiki button in your repository to start creating wiki pages. You have the option of editing in Markdown or one of the other eight editing styles. Wiki pages, like all other material, are saved in Git repositories. Anyone can edit your wiki by default, but you can modify the settings to make it read-only. If your wiki is open to the public, you should keep an eye on community contributions.

C. GitHub Pages:

GitHub Pages is another option for creating documentation for your project repository. These are GitHub-hosted and published webpages. They're written in Markdown, and you can customise the style with GitHub-provided themes. Your Google Analytics tracking ID can also be added to each of your Pages. You can publish once you're happy with your content and the theme.

Your GitHub pages will have the following URL by default: [http://\[accountName\].github.io/\[repoName\]](http://[accountName].github.io/[repoName]). Even if your repository is private, GitHub Pages are always public. You can also establish GitHub Pages for yourself or your company.

IV. MANAGING ISSUES IN GITHUB

Using GitHub's issue tracking, you may track documentation issues (bugs and feature requests) in a project. Many of the features will be familiar if you've used previous issue tracking systems, and like with any system, you'll need to consider how granular issues should be, how you want to identify them, and what best practises your team should follow. You may use Markdown to add formatting, make task lists to track progress, apply emoticons, and more to GitHub issues.

Check out ZenHub, a productivity application that can be integrated with GitHub, to view and manage your issues on a virtual task board (which is perfect for agile shops and virtual teams).

V. USING GITHUB FOR VERSION CONTROL

At its essence, GitHub is a version control system. Files are saved in Git repositories, and knowing how to utilise Git for version control will come in handy while dealing with GitHub. Here's a fast reference to Git commands: <https://git-scm.com/docs>.

Because documentation files are stored with code and other project assets in GitHub, they are not isolated and follow the same workflow as coding. The project's collaborators can review and edit documentation files on GitHub.

VI. INTEGRATIONS WITH OTHER TOOLS AND SUPPORT FOR STUDENTS

Slack, ZenHub, and Travis CI are just a few of the productivity tools that can be linked with GitHub. See <https://github.com/integrations> for a complete list. See <https://github.com/integrations/gitbook> for further information on how to utilise GitBook to host and write documentation. <https://education.github.com/pack/github->

Software development is a collaborative activity that takes place in a group setting. It is not enough to simply place people in teams and assume that everyone understands or agrees on what to do; the implementation of team structures is not simple and does not always result in success; it is not enough to simply put people in teams and assume that everyone knows or agrees on what to do. Each developer in such a team works on the same source code but on distinct jobs. Someone may be updating the front-end, while another is working on the back-end, but every change made by one developer will effect all the others. When multiple developers are working on the same code at the same time, git is essential.

Without version control, each developer would rewrite the work of another. Git allows contributors to make changes without interfering with each other's work. However, it is required to reconcile modifications made by developers in the same piece of code, which can be done via a pull request. A pull request can be thought of as a response to the question, "What has changed in the code?" Rather of merging the changes into the main branch, a pull request allows for "receiving feedback" on the change and improving the supplied code (e.g., finding mistakes which the developer who submitted the pull request is not aware of). Pull requests start a conversation between the person who submits the suggested modifications to the codebase and the people who review it.

- [1] marco ortu, giuseppe destefanis , daniel graziotin, michele marchesi, and roberto tonelli, “how do you propose your code changes?empirical analysis of affect metrics of pull requests on github, , Received May 13, 2020, accepted June 10, 2020, date of publication June 15, 2020, date of current version June 25, 2020.
- [2] M. Wyrich and J. Bogner, “Towards an autonomous Bot for automatic source code refactoring,” in Proc. IEEE/ACM 1st Int. Workshop Bots Softw. Eng. (BotSE), May 2019, pp. 24–28.
- [3] Y. Yu, H. Wang, V. Filkov, P. Devanbu, and B. Vasilescu, “Wait for it: Determinants of pull request evaluation latency on GitHub,” in Proc. IEEE/ACM 12th Working Conf. Mining Softw. Repositories, May 2015, pp. 367–371.
- [5] P. Bell and B. Beer. Introducing GitHub. Sebastopol, CA: O’Reilly Media, 2014.
- [6] GitHub Glossary. (GitHub Help) [Online]. Available: <https://help.github.com/articles/github-glossary/>
- [7] Monika D.Rokade, Dr.Yogesh kumar Sharma,”Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic.” IOSR Journal of Engineering (IOSR JEN),ISSN (e): 2250-3021, ISSN (p): 2278-8719
- [8] Monika D.Rokade, Dr.Yogesh kumar Sharma”MLIDS: A Machine Learning Approach for Intrusion Detection

- All rights reserved by
- www.ijssrd.com