Collaborating and Contributing in GitHub for Technical Communicators

Nicky Bleiel
Watson Information Developer, IBM
STC Summit 2017
National Harbor, MD
May 7, 2017
What We’ll Learn

• Overview and Tour of GitHub
• Terminology
• Working in GitHub
  – Project Documentation options
  – Issue Tracking
  – Version Control
• Integrations with other tools
GitHub

Web-based repository for software projects

Hosts over 57 million repositories that include code and the documentation for that code.

Octocat, the GitHub Mascot
(Simon Oxley)

https://github.com/
GitHub uses Git.

Distributed version control system for software development

Developed by Linus Torvalds, creator of Linux.

#1 code management tool, adopted by one third of software developers
Launched 2008, why now?

GitHub adds features to Git repositories, such as bug tracking and feature requests.

Simplified the process of contributing to open source projects.

Flickr: [David Hilowitz](https://www.flickr.com/photos/davidhilowitz)
This Is Responsive

Patterns, resources and news for creating responsive web experiences.

- This Is Responsive web site
- Announcement post by Brad Frost
- Contribution guidelines

Flickr: [www.audio-luci-store.it](http://www.audio-luci-store.it)
Individuals with a free account:
• Can create public repositories and contribute to projects.

For a fee:
• Can create private repositories

Organizations can post projects on the public version of GitHub for free, as well as purchase GitHub Enterprise for their internal use.

Flickr: Daniel Oines
Opportunities for TCs

Public repositories – contribute content and comments

Corporate Public Projects – project docs, community management

Corporate GitHub Enterprise Projects (and private repositories) – project docs, issue tracking, project file management

Docs live with the code; reviewed with the code
Exercise 1

HELLO! PART 1 — GET READY
Hello! Part 1 – Get Ready

1. Create an account: https://github.com/
2. Install GitHub Desktop: https://desktop.github.com/
3. Install Atom: https://atom.io/
GitHub Terminology

**Repository:** 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.

**Branch:** A branch is a parallel version of a repository. It is contained within the repository, but does not affect the primary or master branch allowing you to work freely without disrupting the "live" version.

**Fork:** A fork is a personal copy of another user's repository that lives on your account.

From the [GitHub Glossary](https://github.com/glossary) - Flickr: [MarcoG2012](https://www.flickr.com/photos/marco2012/)
**Commit**: A commit, or "revision", is an individual change to a file (or set of files). It's like when you save a file, except with Git, every time you save it creates a unique ID (a.k.a. the "hash") that allows you to keep record of what changes were made when and by who. Commits usually contain a commit message which is a brief description of what changes were made.

**Pull request**: Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators.

From the [GitHub Glossary](https://github.com/glossary)
GitHub Terminology

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

**Collaborator:** A collaborator is a person with read and write access to a repository who has been invited to contribute by the repository owner.

**Contributor:** A contributor is someone who has contributed to a project by having a pull request merged but does not have collaborator access.

From the [GitHub Glossary](https://github.com/glossary)
Exercise 2

HELLO! PART 2 — CREATE A REPO
Hello! Part 2 – Create a Repo w/ReadMe

1. Click + > New Repository
2. Name it **hello-world**
3. Add a description
4. Click the **Initialize this Repository with a README** checkbox
5. Click **Create Repository**
Hello! Part 2 – continued

1. Click on ReadMe.md and click the Edit this file button.
2. Copy content from ReadMe_Markdown_Sample.txt
3. Paste it into ReadMe.md
4. Check Preview Changes
5. Scroll down to Commit changes.
6. Choose Commit directly to the master branch.
7. Click Commit Changes.
GitHub Documentation

Options

Readmes

GitHub Pages

Wikis

Real Color Wheel

realcolorwheel.com
Readme

In GitHub repositories, the Readme file is the home page.

Authored in Markdown, and has the file extension of .md.

README checklist by Daniel Beck: https://github.com/ddbeck/readme-checklist
Lightweight markup language that can be converted to HTML easily.
Originally developed in 2004 by John Gruber, and has splintered into different variations. GitHub uses "GitHub Flavored Markdown"
Extra features: syntax highlighting, task lists, and tables.

Fun stuff: Emoji Cheat Sheet
Useful: Markdown Cheat Sheet
Wikis

Every repository can have one. Author in Markdown, or one of the other 8 edit modes.

Wiki pages are stored in Git repositories like all other content.

By default, anyone can edit your wiki, but you can make your wiki read-only.

https://github.com/showcases/projects-with-great-wikis
Exercise 3

HELLO! PART 3 — THE WIKI
Hello! Part 3 – The Wiki

1. Click the **Wiki** tab.
2. Click **Create the first page**.
3. Name the Page “At a Glance: STC Facts”
4. Copy the content from **Wiki_Sample.txt** and paste it in the body.
5. Click **Save page**.
GitHub Pages

Webpages hosted and published on GitHub.

Authored in Markdown

GitHub provides themes to create a custom look.

You can add your Google Analytics tracking ID to each of your Pages.

By default, the URL of your GitHub pages will be: http://[accountName].github.io/[repoName].

GitHub Pages are always public, even if your repository is private.

https://github.com/showcases/github-pages-examples
Reuse

Share snippets using **Gists**: https://gist.github.com

Can be public or “secret”

You can download or embed a Gist

Flickr: [Chris Potter](http://www.flickr.com/photos/chrisp)
ISSUE TRACKING
Decisions, Decisions

Need to decide:

• How granular issues should be
• How you want to label them
• Best practices

GitHub issues can be authored in Markdown, so you can add formatting, create task lists to track progress, use emojis, and more.
You can ...

• Create **Labels** and **Milestones**
• Tag every issue with one or more labels, a milestone, and an **assignee** (owner).
• Find issues using Filters and Search, or click on any label or milestone
• Create **Epics** and assign issues to them.
Drawbacks

• Searching across repositories: https://help.github.com/articles/searching-issues/

• Gathering statistics across repositories

• Permissions: https://help.github.com/articles/access-permissions-on-github/
Hello! Part 4 – Let’s Collaborate
Handy Features

• To reference another issue, enter “#”
The chosen issue will become a link

• Anyone with Collaborator status can comment. Enter “@” to ping another collaborator.

• Task lists

• If you believe something is ready to ship, mark it with the “Ship It Squirrel” emoji (:shipit:)
ZenHub

https://www.zenhub.io/
VERSION CONTROL
Workflow

• There are many options, many tools
• You can work in the GitHub GUI, strictly on the command-line using Git commands, with a mixture of both, or with an app.
• If using GitHub Enterprise, your company may develop guidelines you need to follow around the naming of branches and who should do reviews and merges — or you may develop them.
• Check out: https://help.github.com/articles/what-is-a-good-git-workflow/
Docs follow the same workflow as code. All project collaborators can review and contribute.
CREATE A BRANCH
Create a branch in your project where you can safely experiment and make changes.

ADD COMMITS

OPEN A PULL REQUEST
Use a pull request to get feedback on your changes from people down the hall or ten time zones away.

DISCUSS AND REVIEW

MERGE AND DEPLOY
Merge your changes into your master branch and deploy your code.

https://guides.github.com/introduction/flow/
and https://guides.github.com/pdfs/githubflow-online.pdf
Anything in the master branch is always deployable.
File Facts

• Any file that can be read with a text editor can be opened and edited within GitHub.
• .gitignore (local stuff, etc. you don’t want in repo)
• Even though you can open many file types (such as .dita files) in GitHub, you may prefer to use your XML editing tool.
• Binary files (images, Word files, etc.) can’t be opened within GitHub at all.
  – This is part of the reason why working on a local clone is a best practice. On your local copy, you can open any file in the application you wish.
Storing Binary files in Git

• This blog post gives a good overview of why storing binary files in Git can be an issue: https://robinwinslow.uk/2013/06/11/dont-ever-commit-binary-files-to-git/

• Git LFS is one possible solution: https://git-lfs.github.com/; git-annex is another: https://git-annex.branchable.com/

• Or create a custom solution or separate documentation repo.
Exercises 5 & 6

HELLO! PARTS 5&6 — WORKFLOWS
Hello! Part 5 – GitHub Workflow

1. Open **Readme.md** and click the **Edit this file** button. Make a change. (Lorem Ipsum)
2. Scroll down to **Commit changes**.
3. Choose **Create a new branch for this commit and start a pull request**.
   Name the branch “Your name –**NewReadMe**”
4. Enter a description of what you did.
5. Click **Propose New File Change**.
Hello! Part 5 - continued

1. Check the diff.
2. Add a description/instruction.
3. Click **Create Pull Request**.

- Explore -

1. Go back to PR
2. Click **Merge Pull Request**
3. Click **Delete Branch**
Hello! Part 6 – GitHub Desktop

1. Create a new repo. Initialize w/ReadMe and add a license.

2. Click **Upload Files**. Upload the files in the “GitHub Desktop Exercise” folder.

3. Commit to the Master branch.

4. Create a new branch.

5. Click **Clone or Download**, choose **Open in Desktop**. Choose a folder on your machine – the repo will be cloned there.
Hello! Part 6 – continued

1. In GitHub Desktop, switch to the new branch.
2. Open your local repo in Explorer.
3. Open ReadMe.md from Explorer. Make a change (add a release note) and save the file.
4. Add a message and commit the changed file to the new branch.
5. Click **Pull Request** (upper right). Create a PR.
6. Go to GitHub. Check the PR and merge it.
7. Go to GitHub Desktop and sync.

Note: You can also clone a fork.
INTEGRATIONS WITH GITHUB
Integrations with GitHub

- Almost 150 productivity tools can be integrated with GitHub, including Slack and ZenHub. See [https://github.com/integrations](https://github.com/integrations).

- GitBook can be used to host and write books, see [https://github.com/integrations/gitbook](https://github.com/integrations/gitbook).
SOCIAL FEATURES
“Follow” people

“Watch” projects

“Star” projects & visit your “Stars” page

Flickr: Antonio Silveira
Exercise 7

HELLO! PART 7 — BEING SOCIAL
Hello! Part 7 – Being Social

"Watch" or “Star” projects
1. Profile Photo > "Explore"
2. Find a project
3. Click the "Watch" or “Star” button at the top
Watch = You'll receive notices about activity (Web and/or Email, Profile Photo > Settings > Notifications)
Star = No notices, you need to check

"Follow" people
1. Search > Advanced Search > Users Options
2. Find someone you know, or whose work you admire
3. Click the "Follow" button
You will receive notices about their activity

To check "Stars“ (and more), go to Profile Photo > Your Profile
TECH COMM PROJECTS ON GITHUB
Dynamic Information Model (DIM)

https://github.com/oxygenxml/dim

Open source (Apache 2.0 license)

Contributors

- oXygen XML Editor
- Comtech Services
Projects of Interest

• **Dynamic Information Model** (an implementation of an intelligent style guide) by oXygen xml editor and Comtech Services: [https://github.com/oxygenxml/dim](https://github.com/oxygenxml/dim).

• **W3C HTML Specification**: [https://github.com/w3c/html](https://github.com/w3c/html)

• **DITA Open Toolkit**: [https://github.com/dita-ot/dita-ot](https://github.com/dita-ot/dita-ot)


GitHub is now being used to collaborate on projects as diverse as Gregorian chants, licensing agreements, and wedding invitations

From Collaborative Coding to Wedding Invitations: GitHub Is Going Mainstream

Flickr: Lars Plougmann
Contact information:
Nicky Bleiel
nableiel@us.ibm.com
nickybleiel.com
Git Resources

List of Git commands: https://git-scm.com/docs


Visual Git Cheat Sheet: http://ndpsoftware.com/git-cheatsheet.html

Git for Humans slides: https://speakerdeck.com/alicebartlett/git-for-humans

Git for Humans book: https://abookapart.com/products/git-for-humans