Collaborating and Contributing in GitHub for Technical Communicators

Nicky Bleiel
Watson Information Developer, IBM
tcworld conference 2017
Stuttgart, Germany
24 October, 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, code, and comments

Corporate Public Projects – project docs, code, 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

Ordered Listocat by Cameron McEfee
Hello! Part 1 – Get Ready

• Create a free GitHub account: https://github.com/
• Install Atom (Markdown editor): https://atom.io/
• Install GIT https://git-scm.com/downloads
• Install GitHub Desktop (A GUI for GIT) : https://desktop.github.com/
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://flickr.com/photos/marco2012)
GitHub Terminology

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
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/guides/glossary)
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.

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

https://github.com/showcases/projects-with-great-wikis
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
## Issue Tracking

<table>
<thead>
<tr>
<th>#</th>
<th>Issue Title</th>
<th>Status</th>
<th>Created By</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Velocity</td>
<td>Open</td>
<td>LouisTCat</td>
<td>Apr 24, 2016</td>
</tr>
<tr>
<td>2</td>
<td>New</td>
<td>Open</td>
<td>LouisTCat</td>
<td>May 3, 2016</td>
</tr>
<tr>
<td>3</td>
<td>Evaluate backlog</td>
<td>Open</td>
<td>LouisTCat</td>
<td>Apr 24, 2016</td>
</tr>
<tr>
<td>4</td>
<td>Update the wiki</td>
<td>Open</td>
<td>LouisTCat</td>
<td>May 1, 2016</td>
</tr>
<tr>
<td>5</td>
<td>Write tests</td>
<td>Open</td>
<td>LouisTCat</td>
<td>Apr 24, 2016</td>
</tr>
<tr>
<td>6</td>
<td>Fix This Bug</td>
<td>Open</td>
<td>LouisTCat</td>
<td>May 1, 2016</td>
</tr>
<tr>
<td>7</td>
<td>A feature to consider</td>
<td>Open</td>
<td>LouisTCat</td>
<td>Apr 24, 2016</td>
</tr>
<tr>
<td>8</td>
<td>Add this cool feature</td>
<td>Open</td>
<td>LouisTCat</td>
<td>Apr 24, 2016</td>
</tr>
<tr>
<td>9</td>
<td>So close to getting this done</td>
<td>Open</td>
<td>LouisTCat</td>
<td>Apr 24, 2016</td>
</tr>
<tr>
<td>10</td>
<td>Need to get this done</td>
<td>Open</td>
<td>LouisTCat</td>
<td>Apr 24, 2016</td>
</tr>
<tr>
<td>11</td>
<td>Add Important Information</td>
<td>Open</td>
<td>LouisTCat</td>
<td>Apr 24, 2016</td>
</tr>
</tbody>
</table>
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.
Collaborating with your Team
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/
Team collaboration tools

• Searching across repositories:

• Permissions:
  https://help.github.com/articles/access-permissions-on-github/
VERSION CONTROL
This is Git. It tracks collaborative work on projects through a beautiful distributed graph theory tree model.

Cool. How do we use it?

No idea. Just memorize these shell commands and type them to sync up. If you get errors, save your work elsewhere, delete the project, and download a fresh copy.
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
GitHub’s Golden Rule

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.
INTEGRATIONS WITH GITHUB
Integrations with GitHub

• Almost 150 productivity tools can be integrated with GitHub, including Slack and ZenHub. See https://github.com/marketplace.

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

“Watch” projects

“Star” projects & visit your “Stars” page

Flickr: Antonio Silveira
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)
But it is not just for code ...

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
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
Your opinion is important to us! Please tell us what you thought of the lecture. We look forward to your feedback via smartphone or tablet under http://ptk07.honestly.de

or scan the QR code

The feedback tool will be available even after the conference!