

APPENDIX A: SERVICES

1. Train the Trainer

This four (4) days course will prepare selected individuals on your team to deliver the GitHub training course.

2. Pre-Work

Prior to attending the course, each participant will be required to complete a pre-test to assess their current Git and GitHub proficiency. This pre-test is used to assess the amount of review of core concepts that will be required during the course.

3. Training Preparation

Day one (1) through three (3) includes trainer-candidate preparation and material walkthrough. The delivery of the course is tailored to fit the needs of a future trainer by going over:

- Completing a detailed walkthrough of the GitHub for Developers flow and activities
- Digging deeper into the Git, GitHub, and workflow concepts discussed during the course
- Preparing to handle frequently asked questions and common problem areas for learners
- Troubleshooting strategies for common issues encountered in networked operations
- Preparing to lead workflow discussions
- Accessing extended modules in response to user questions and workflow needs

1.1 Getting Started with Collaboration

- Introducing Git and GitHub
 - Key Features
 - The Ecosystem
- Exploring a repository
- Using Issues

1.2 Understanding the GitHub Flow

- Overview of the GitHub Flow
- Creating and using branches on GitHub

1.3 Working Locally

- Basic Command Line Interface (CLI) configuration
 - Understanding Git configuration levels
- Cloning a Repository
 - Understanding the distributed nature of Git
 - Why we clone

1.4 Creating and Committing Local Files

- Creating a local file

- Understanding the Two Stage Commit
- Pushing your changes to GitHub
- Creating a Pull Request
 - Using GitHub flavored markdown
 - Special keywords for cross-linking Issues and Pull Requests
 - Commenting on Pull Requests and using code review
- Editing files via the Pull Request
- Merging Pull Requests
 - Social conventions for merging
 - What happens when we merge?
- Cleaning up local and remote tracking branches

1.5 Understanding Workflow Variations

- Working on GitHub.com
- Working on both GitHub.com & locally
- Working locally

1.6 Viewing Local Diffs and History

- Viewing local diffs
- Viewing project history

1.7 Creating Shortcuts

- Creating aliases

1.8 The Workflow End-to-End

- Forking a repository
- Cloning the repository
- Creating local branches
- Workflow review for a simple file change
- Cleaning up local and remote tracking branches

1.9 Fixing Common Issues with Git

- Handling merge conflicts locally
- Handling merge conflicts from the remote
- Reverting commits
- Creating atomic commits
 - Staging hunks of changes
- Moving and renaming files with Git

1.10 Creating a Repository Locally

- Creating a Repository on the command line

1.11 Rewriting History

- How Git stores data (and why you should care)

- Amending your last commit
- Understanding Git reset
- Using the reflog
- Cherry-picking a single commit
- Using rebase and rebase interactive

1.12 Workflow Application

- Using what you have learned in your day-to-day work
- Discussing workflows and best practices

1.13 Extended topics

In the case of a fast-moving class or extra time, these topics may be covered to provide additional mastery of Git and GitHub:

- .gitignore
- Bisecting
- Managing multiple remotes
- Tags & releases
- Submodules, subtrees and dependency management
- Other customer requests

4. Hands-on Practice

After we have surveyed the curriculum, we will give future trainers the opportunity to demonstrate mastery of the concepts and practice teaching and answering questions.

Teachbacks:

On day four (4), participants will pair up with their classmates and teach selected topics within the curriculum. Each participant will receive feedback from peers as well as the instructor on their practiced delivery.