

**Git and GitLab Fundamentals for Engineers**

**Course Number:** GIT-114  
**Duration:** 2 days

**Overview**

This Git and GitLab Fundamentals for Engineers training course teaches attendees how to use Git at a fundamental level and includes plenty of hands-on exercises managing repositories, creating and merging branches, properly backing out of mistakes, and getting comfortable resolving conflicts. This course gives students a solid grasp of how Git works, helping reduce the typical “fear of blowing something up” in a Git project. Students consistently complete the class with a very high level of confidence with Git.

This course leverages GitLab as a platform to host our Git repositories. Students learn how to share changes, synchronize their branches, collaborate through merge requests, review code, and other collaboration best practices on GitLab.

**Prerequisites**

* No prior experience with Git is presumed.
* Prior experience with other version control systems is helpful but not required.
* Experience with the command-line or DOS command prompt is preferred.

**Materials**

All Git training attendees receive comprehensive course materials in digital format.

**Software Needed on Each Student PC**

* Git 2.x or later
* Internet access for all attendees and the instructor

**Objectives**

All students will:

* Understand Git and Git fundamentals
* Review and edit the commit history
* Improve your daily workflow
* Branch, merge, and use remote repositories
* Understand collaboration best practices as a team
* Understand the GitLab flow

**Outline**

* Introduction to Source Code Management
  + The Core Principles of Change Management
  + The Power to Undo Changes
  + Audit Trails and Investigations
  + Reproducible Software
* Git Introduction and Basics
  + Introduction to Git
  + Trees and Commits
  + Configuring Git
  + Adding, Renaming, and Removing Files
* Reviewing and Editing the Commit History
  + Reviewing the Commit History
  + Revision Shortcuts
  + Fixing Mistakes
* Improving Your Daily Workflow
  + Simplifying Common Commands with Aliases
  + Ignoring Build Artifacts
  + Saving Changes for Later Use (Stashing)
* Branching
  + Branching Basics
  + Listing Differences Between Branches
  + Visualizing Branches
  + Deleting Branches
  + Tagging
* Merging
  + Merging Basics
  + Merge Conflicts
  + Merging Remote Branches
* Remote Repositories (i.e., GitLab)
  + Remote Repositories
  + Synchronizing Objects with Remotes
  + Tracking Branches
  + Remote branch management
  + Forking and working with upstreams
  + Rolling back changes
* Collaboration Through a Platform
  + Introduction to GitLab
  + Git Repositories on GitLab
  + Daily Workflow - The GitLab Flow
  + Using pull requests and code reviews
  + Leveraging Forks
* History Management
  + Rebasing
  + Searching the log
  + Cherry picking
  + Squashing
  + Bisecting
  + Advanced workflows
* Workflows/Best Practices
  + Branch strategies
  + Remote strategies
  + Tagging strategies
* Repairing Branches Mastering History
  + Reset
  + Reflog
* Conclusion