

# Git and GitHub for engineers

A 1 day **Hands on** training course



## Description

This course covers version control using Git but also using GUI frontends such as GitHub. The course starts with a tour of using GitHub but then quickly moves onto using git from the command line. All elements of git version control are covered including creation of repositories, adding and editing files, branches and merging, rewriting history and handling merge conflicts. Hands on sessions are used throughout the course.



## Key outcomes

By the end of the course delegates will be able to:

- ✓ Install git.
- ✓ Perform version control using git and GitHub.
- ✓ Add and edit files in a repository.
- ✓ Create branches and perform merges..
- ✓ Handle merge conflicts.



## Training approach

This structured course uses Instructor Led Training to provide the best possible learning experience. Small class sizes ensure students benefit from our engaging and interactive style of teaching with delegates encouraged to ask questions throughout the course. Quizzes follow each major section allowing checking of learning. Hands on sessions are used throughout to allow delegates to consolidate their new skills.



## Details

### Who will benefit?

Anyone requiring version control.

### Prerequisites

None.

**Duration:** 1 day

**Customer rating:**

**New course**

### Generic training



Generic training compliments product specific courses covering the complete picture of all relevant devices including the protocols "on the wire".

*"Friendly environment with expert teaching that teaches the why before the how."*  
G.C. Fasthosts

### Small class sizes



We limit our maximum class size to 8 delegates; often we have less than this. This ensures optimal interactivity between delegates and instructor.

*"Excellent course. The small class size was a great benefit..."*  
M.B. IBM

### Hands On training



The majority of our courses use hands on sessions to reinforce the theory.

*"Not many courses have practice added to it. Normally just the theoretical stuff is covered."*  
J.W. Vodafone

### Our courseware



We write our own courses; courseware does not just consist of slides and our slides are diagrams not bullet point text.

*"Comprehensive materials that made the course easy to follow and will be used as a reference point."*  
V.B. Rockwell Collins

### Customise your course



Please contact us if you would like a course to be customised to meet your specific requirements. Have the course your way.

*"I was very impressed by the combination of practical and theory. Very informative. Friendly approachable environment, lots of hands on."*  
S.R. Qinetiq

# Git and GitHub for engineers

## Course content

### Introduction

Version control for software, configuration management. Other uses. Version control systems. What is git? What is GitHub? Distributed version control. Comparison of git to other systems.

### GitHub

Getting started, creating an account, account types, repositories, access control, bug tracking, feature requests. Alternatives to GitHub. Hands on: Using GitHub.

### Installing git

Linux install, Windows install, `git config`, levels, `user.name`, `user.email`. Hands on: Installing and configuring git.

### Creating repositories

`git clone`, `github`, `git remote`, `git init`. Hands on: Creating a repository.

### Adding and editing files

Staging and adding, `git add`, `git commit`, `git push`, `git pull`, `git status`, `git log`. Two stage process. File states: Working, staging, history, untracked. `git mv`, `git rm`, `.gitignore`, `git diff`, `git difftool`. Undoing changes. Hands on: Adding and editing files in git.

### Branching and merging

What is a branch, HEAD label, master branch, `git branch`, `git checkout`. Feature branches, bug fix branches, integration branches, production branches, fast forward merges, 3 way merges, `git merge`, `git status`, `git log`, tags. Hands on: Making branches, merging.

### Rewriting history

`git reset`, `git rebase`, advantages. Hands on: Reset commits, rebase a branch.

### Merge conflicts

What is a conflict, conflict resolution process, resolving merges, rebasing, `git log`, merge tools, configuring merge tools, avoiding conflicts. Hands on: Merge resolution.

