Fontys Git Manual



Last version: 22-11-2020

Authors:  
Tim de Laat  
Kristian Snel

Contents

[What is Git and why should you use it? 3](#_Toc56933765)

[Manual structure 3](#_Toc56933766)

[Git Basic 4](#_Toc56933767)

[Getting started with 4](#_Toc56933768)

[Installation 4](#_Toc56933769)

[Setting up a test project 7](#_Toc56933770)

[Cloning an existing repository 7](#_Toc56933771)

[Setting up a new repository 7](#_Toc56933772)

[Git workflow 7](#_Toc56933773)

[Push pull requests 7](#_Toc56933774)

[Git Technical 9](#_Toc56933775)

[Git Management 9](#_Toc56933776)

[Git Advanced 9](#_Toc56933777)

[Extra info 10](#_Toc56933778)

[Git commands 10](#_Toc56933779)

# What is Git and why should you use it?

Large projects may require a lot of different people to work on a lot of files. Project structure may get cluttered up and files get lost. Git allows a team to keep a clear overview of the project files and the changes made to these files. Git is a so-called distributed version control system. Using a version control system allows a team to keep a log of changes made to the files in the past. Therefore, allowing every member of the team to see how the document progresses.

**Did you know?**Git was created in 2005 by Linus Torvalds. Torvalds is also famous for his creation of the Linux Kernel.

Git is not only useful for teams but also for individuals. A project may span a long amount of time. It can get difficult to keep track of project files and file history. Git provides a set of tools to see previous project versions, or test extra functionality parallel to the original project ensuring the safety of the original project.

This document features as a guideline to Git and how you as a Fontys student can incorporate Git in your standard workflow to prevent issues such as data loss and miscommunication and in general help with maintaining an organized project structure.

# Manual structure

This guide will be split into 4 different tutorials namely:

|  |  |
| --- | --- |
| Tutorial | Aimed at |
| Git basics | Everyone |
| Git technical | Project members |
| Git management | Project management |
| Git advanced | Those who want to use Git to its maximum potential |

To improve the way of working within EXPO groups all disciplines are required to take knowledge of Git. Tips only aimed at a specific discipline will from now on be color coded in the following representative colors:

**ME**

Git provides excellent tools to work and collaborate on CAD files such as Solidworks and Siemens NX.

Mechanical Engineering

Mechatronics engineering

**MT**

Combining electrical, mechanical and software projects? Git is a must know for Mechatronics engineers.

Electrical engineering

**EE**

Working on PLC software or circuit diagrams? Git provides version control to save file history.

# Git Basic

Git also support a CLI  
For more information:   
https://git-scm.com/downloads

**MT**

**EE**

## Getting started with

### Installation

Git supports a great variety of Desktop clients. One of the most popular interfaces is Github Desktop.

Github desktop can be downloaded using this link: <https://desktop.github.com/>



Run the downloaded GitHubDesktopSetup.exe

Sign in to or create a free account.





Log in to your account.



### Setting up a test project



### Cloning an existing repository



### Setting up a new repository



## Git workflow

### Push pull requests

# Git Technical

# Git Management

# Git Advanced

# Extra info

## Git commands

git init

git clone git pull

git config --global user.name git config --global user.email

git touch documentname

git add wildcard \*.html all .

git status

git commit -m 'description what changed' (Waar is die -m ook al weer voor?)

git push

Branching

git branch branchname

git checkin git checkout

gitmerge branchname 'describe merge'