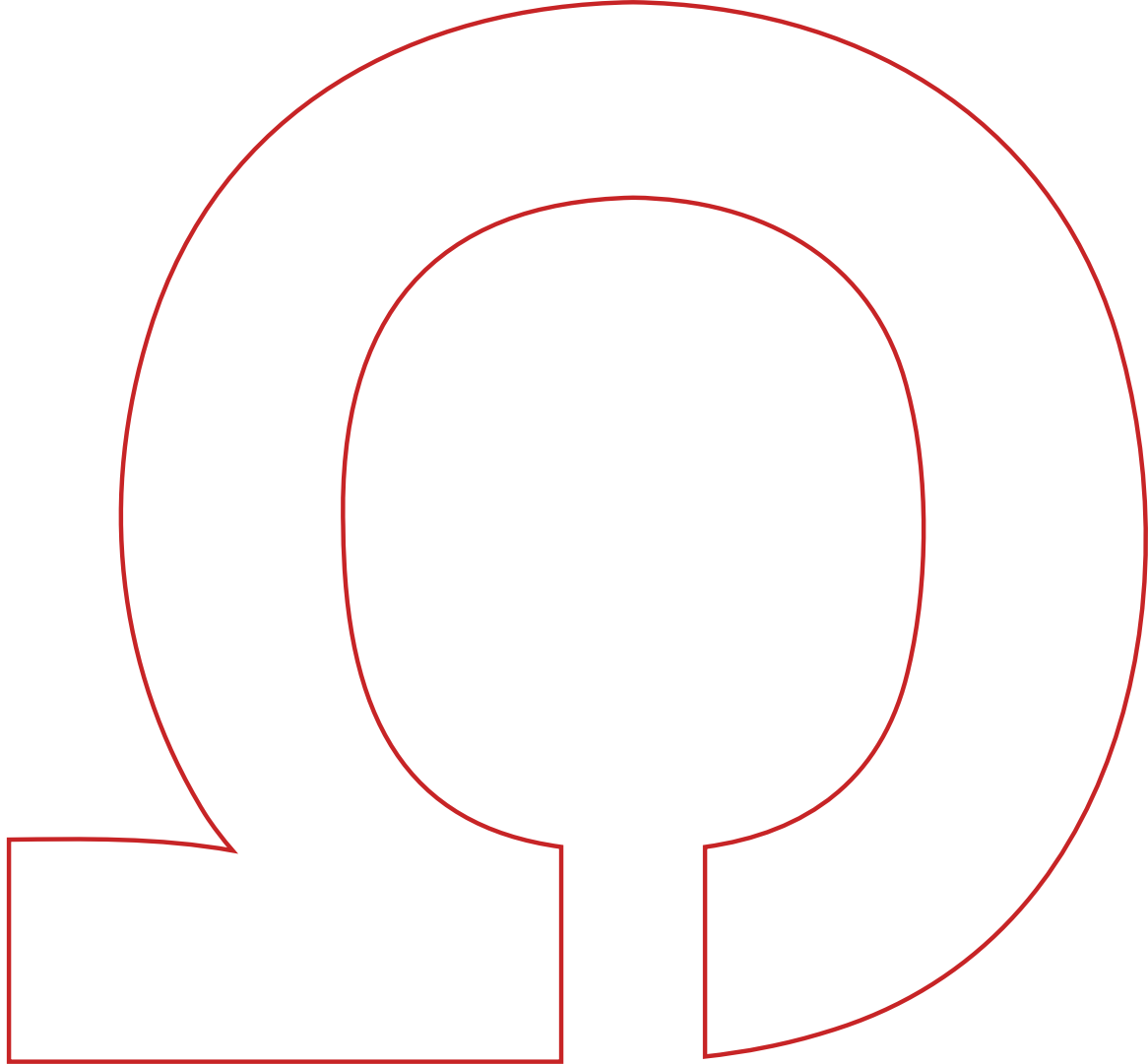


Einführung in Git

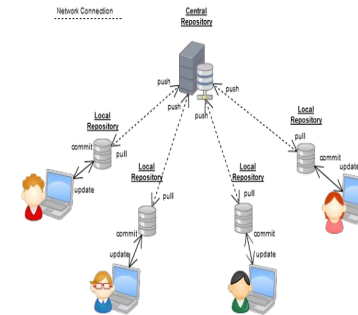


Einführung in Versionsverwaltung mit git

Versionskontrollsysteme (VCS)

Definition

- vereinfacht zu verwaltende Zusammenarbeit bei einem Projekt ohne dass Dateien direkt ausgetauscht/versendet werden müssen
- Möglichkeit, unbegrenzt viele Leute gleichzeitig an derselben Codebasis mitarbeiten zu lassen
- Nachverfolgen von Änderungen möglich
- Einfach, Dateien zurückzusetzen, falls etwas schief lief
- Zwei Varianten:
 - zentralisierte VCS
 - verteilte VCS



Einführung in Versionsverwaltung mit git

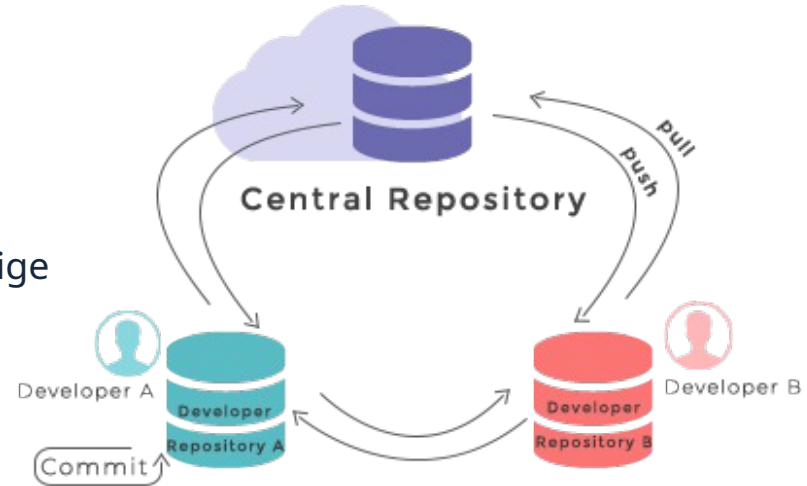
- freies, open-source verteiltes System, bei dem der Fokus auf Geschwindigkeit und Datenintegrität liegt
- initial entwickelt 2005, vom Erfinder von Linux (Linus Torvalds)
- Keine zentralisierte Verbindung notwendig
- Sehr schwer, Code zu verlieren
- möglich, mit sehr vielen Leuten gleichzeitig am selben Projekt zu arbeiten



Einführung in Versionsverwaltung mit git

Grundlegende Funktionsweise von git

- Ein Git repository ist eine Datenbank, die alle Informationen zur Speicherung und Verwaltung eines Projekts, inkl Revisionen und Verlauf, enthält
- jedes Projekt hat ein eigenes Repository
- Repositories werden auf (zentralem) Server gespeichert
- jeder Nutzer hat eine lokale, eigene, unabhängige Kopie eines oder mehrerer Repositories

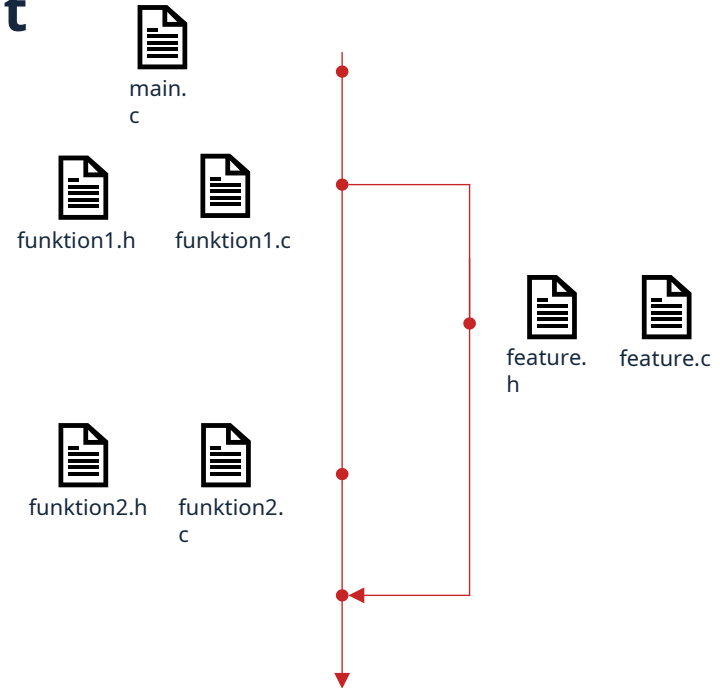


Einführung in Versionsverwaltung mit git

Grundlegende Funktionsweise von git

Beispiel: (ohne git)

- Kim hat Idee für tolle Software, schreibt Datei main.c
- Kim erweitert Programm um Datei funktion1.h, funktion1.c
- Kim bekommt Hilfe von Alex. Alex baut ein Feature, das auf funktion1 aufbaut
- Währenddessen fügt Kim dem Programm noch eine Funktion funktion2.c, funktion2.h hinzu
- Alex ist fertig und möchte den Code zum aktuellen Stand hinzufügen

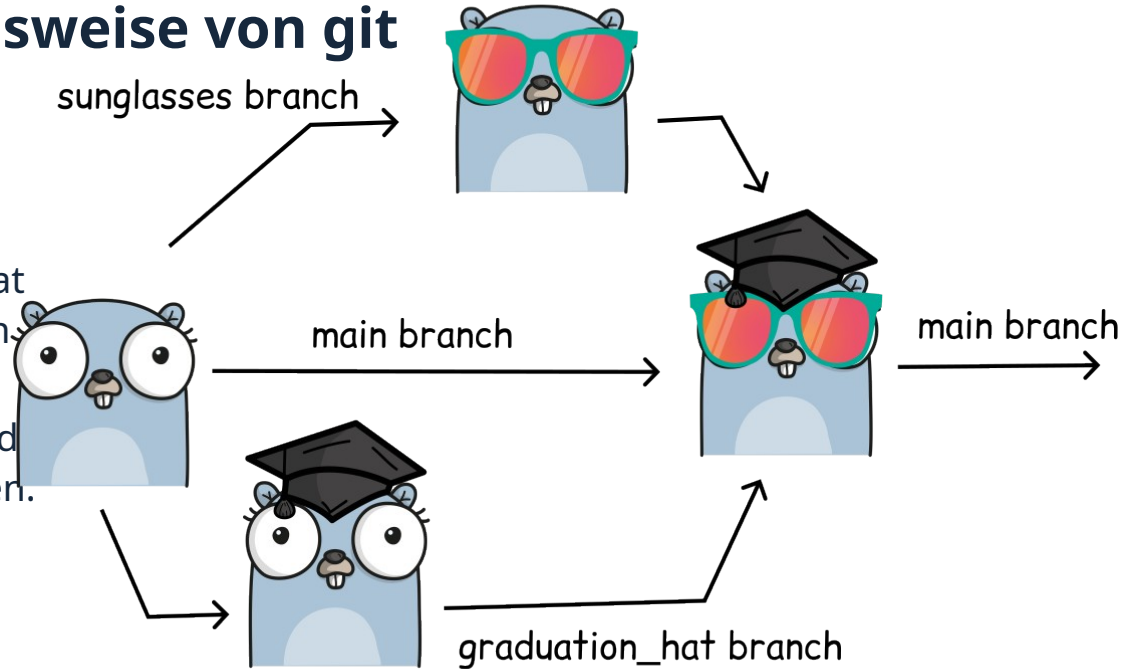


Einführung in Versionsverwaltung mit git

Grundlegende Funktionsweise von git

Beispiel:

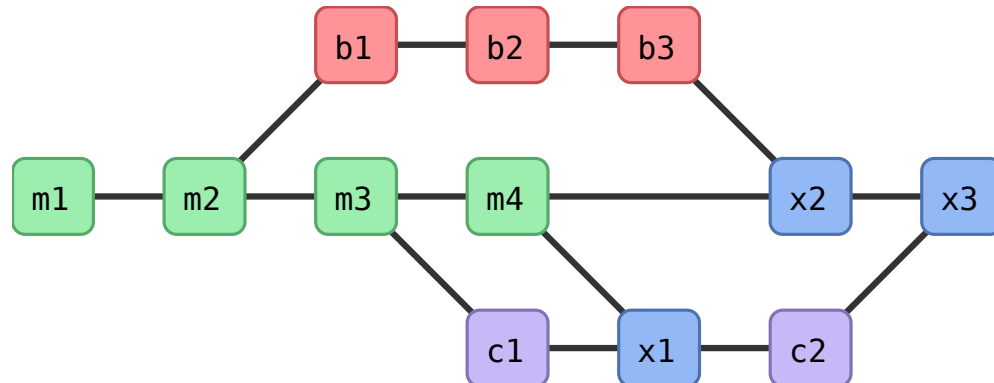
- Der main branch bleibt stabil
- Sunglass feature und graduation_hat fetature können auf ihrem jeweiligen Feature branch entwickelt werden.
- Und können wenn sie fertig sind in d main branch zurück gemergt werden.



Einführung in Versionsverwaltung mit git

Grundlegende Funktionsweise von git

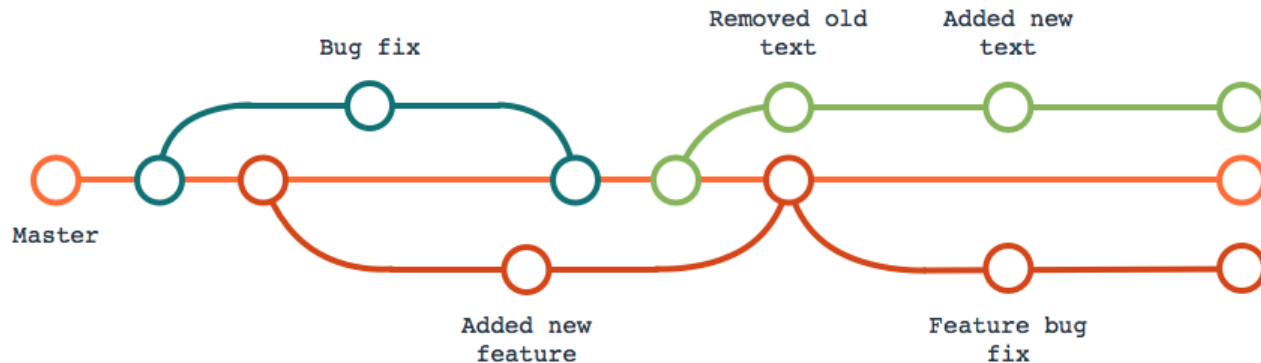
- Baumartige Struktur, mit Zweigen (sog. Branches)
- Beim Starten eines Projekts wird default branch angelegt, meist main oder master genannt
- neue Features oder Bugfixes werden entwickelt, indem ein Branch von master abgezweigt wird
- fertige Features oder Bugfixes werden über sog. merges zum master hinzugefügt



Einführung in Versionsverwaltung mit git

Grundlegende Funktionsweise von git

- Baumartige Struktur, mit Zweigen (sog. Branches)
- Beim Starten eines Projekts wird default branch angelegt, meist main oder master genannt
- neue Features oder Bugfixes werden entwickelt, indem ein Branch von master abgezweigt wird
- fertige Features oder Bugfixes werden über sog. merges zum master hinzugefügt

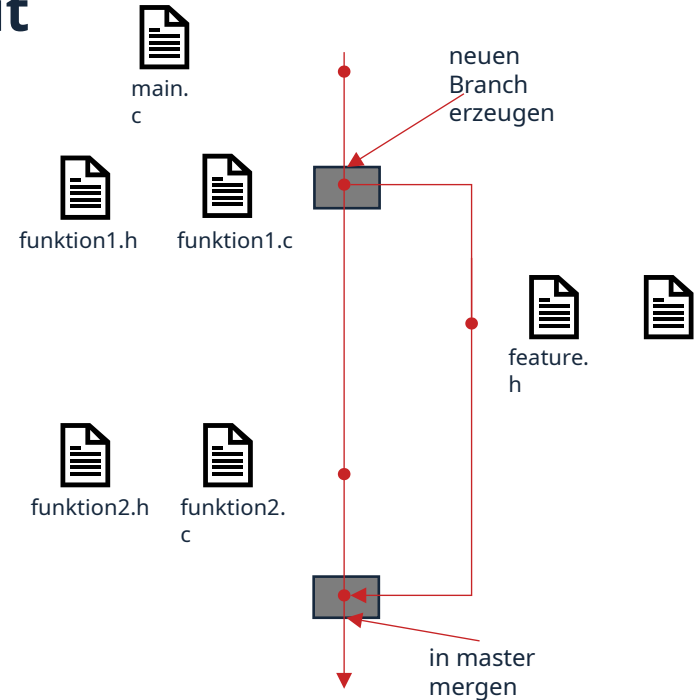


Einführung in Versionsverwaltung mit git

Grundlegende Funktionsweise von git

Beispiel: (mit git)

- Kim hat Idee für tolle Software, schreibt Datei main.c
- Kim erweitert Programm um Datei funktion1.h, funktion1.c
- Kim bekommt Hilfe von Alex. Alex baut ein Feature, das auf funktion1 aufbaut
- Währenddessen fügt Kim dem Programm noch eine Funktion funktion2.c, funktion2.h hinzu
- Alex ist fertig und möchte den Code zum aktuellen Stand hinzufügen



Einführung in Versionsverwaltung mit git

git installieren

Windows/Mac

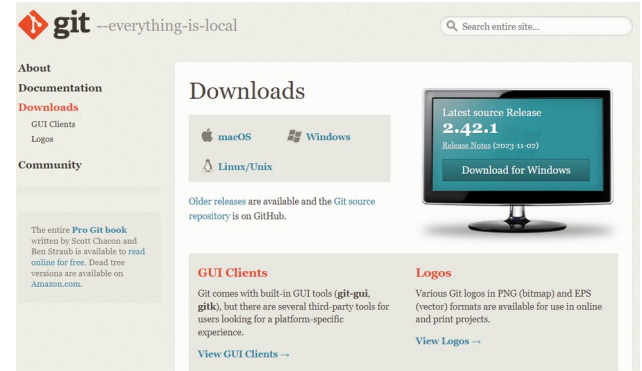
- Aktuelle Version passend zum Betriebssystem herunterladen

<https://git-scm.com/download>

- und installieren

Linux

- Entweder ebenfalls entsprechende Version herunterladen und installieren
- oder per Kommandozeile, z.B. Ubuntu `$sudo apt install git`



Einführung in Versionsverwaltung mit git

Grundlegende Befehle

Befehl	Bedeutung
init	Projekt initialisieren
clone	Projekt vom zentralen Server lokal herunterladen
commit	Aktuelle Version wiederauffindbar speichern (jeder commit hat eindeutigen hash-Wert)
merge	Aktuellen Branch inkl aller mit commit gespeicherten Änderungen in anderen Branch überführen
pull	Neueste Änderungen vom Server holen
push	Aktuellen lokalen Stand auf den Server bringen

Einführung in Versionsverwaltung mit git

Beispiel (mit Konsole)



- Windows: Git Bash starten (z.B. vom Windows Startmenü)
- Unter Ubuntu einfach wie gewohnt weiter auf der Kommandozeile arbeiten und git Befehle nutzen
- mittels cd in passendes Verzeichnis wechseln

- git clone <Repo-URL>
- git commit -m "hier-commit-message-eintippen"
- git push
- git pull

- Mit git Kommandozeile vertraut machen

<http://guides.beanstalkapp.com/version-control/common-git-commands.html>

A screenshot of a Git Bash terminal window. The window title is "MINGW64~/z:/Users/freudenreichan/Lehrveranstaltungen/WiSe2023/Info2_Praktikum/info2". The terminal shows the following commands and their outputs:

```
f@freudenreichan\FH033006 MINGW64 ~
$ cd Lehrveranstaltungen/wiSe2023/Pr
Praxissees-ter/ Programmieren_Informatik.zip
f@freudenreichan\FH033006 MINGW64 ~
$ cd Lehrveranstaltungen/wiSe2023/Info
Info2/ Info2_Praktikum/ Info_Grundlagen/
f@freudenreichan\FH033006 MINGW64 ~
$ cd Lehrveranstaltungen/wiSe2023/Info2_Praktikum/inf
Info2Prac_Testatb.pdf Info2/
f@freudenreichan\FH033006 MINGW64 ~
$ cd Lehrveranstaltungen/wiSe2023/Info2_Praktikum/info2/
f@freudenreichan\FH033008 MINGW64 ~/Lehrveranstaltungen/wiSe2023/Info2_Praktikum/
info2 (master)
$
```

Git: Konsole vs GUI

Warum git mit Kommandozeile nutzen?

- funktioniert gleich auf allen Systemen
- einzige Möglichkeit, um ALLE Git Befehle nutzen zu können
 - GUIs nutzen nur einen Teil der existierenden Git-Befehle
- die meisten Onlinetutorials arbeiten mit Kommandozeilenbefehlen
- Hilfestellungen zu Fragen im Internet meist mittels Kommandozeilenbefehlen beantwortet

Warum git mit GUI nutzen?

- genaue Syntax des Befehls muss nicht bekannt sein
- übersichtlicher und (als Einsteiger) einfacher zu nutzen
- leichter Dateien zu vergleichen (z.B. weil Änderungen vom Server sind anders als eigene, lokale Änderungen)

Github

- öffentliche Website/Server, um Git zu nutzen (github.com)
- entwickelt 2008
- seit 2018 Tochterfirma von Microsoft
- öffentliche und private Git Repos möglich
- Muss man Github nutzen, um Git nutzen zu können? Nein!
 - Github bietet Möglichkeit, mit wenig Aufwand nicht-lokalen Gitserver zu nutzen



Autoren / Impressum

- **Autoren**

Prof. Dr. Anja Freudenreich

Prof. Dr. Jan Paulus

Prof. Dr. Enrico Schröder

- **Impressum**

Prof. Dr. Anja Freudenreich

Fakultät Elektrotechnik Feinwerktechnik Informationstechnik,

Wassertorstraße 10

90489 Nürnberg, Germany

Tel: +49-911-5880-1098

E-mail : anja.freudenreich@th-nuernberg.de

Dieses Skriptum ist nur für den eigenen Gebrauch im Studium gedacht. Eine Weitergabe ist nur mit Zustimmung des Autors gestattet.