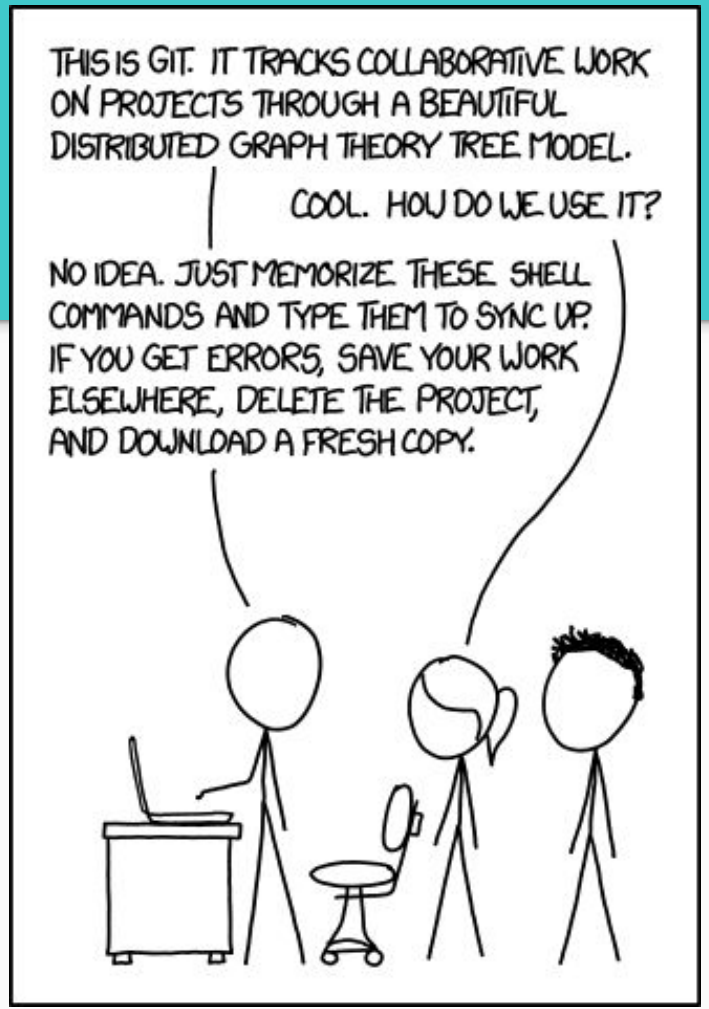
Versionsverwaltung mit git

Edirom Summer School 2021

Git?



https://xkcd.com/1597/

Themen

- Versionsverwaltung
- Git-Grundlagen
- (Forschungs-)Daten in Git
- Andere Git-Projekte nutzen
- Eigenes Git(Hub)-Projekt erstellen und strukturieren
- An Git-Projekten (zusammen) arbeiten
- Weiterführende Möglichkeiten für Git und GitHub

Motivation für Versionsverwaltung?

Motivation für Versionsverwaltung?

- produktive Kollaboration an Daten oder Code
- gezielte Wiederherstellung der Versionshistorie oder Teilen davon
- Nachvollziehbarkeit bei der Erstellung oder Bearbeitung von Daten
- Automatisierung von Qualitätschecks o.Ä. während der Arbeit an Daten

Hürden für Versionsverwaltung?

- höherer Aufwand
- oft steile Lernkurve
- Kollaboration: Alle müssen dasselbe Tool nutzen und über entsprechende Kenntnisse verfügen

Versionsverwaltung

Was ist Versionsverwaltung/-kontrolle?

- Verfolgung von Änderungen durch Versionen

Was ist Versionierung?

- Vergabe von Versionierungsnummern

VCS = Version Control System

Git-Grundlagen: Dokumentation

- Pro Git book, digital kostenfrei verfügbar
- GitHub Git Cheat Sheet

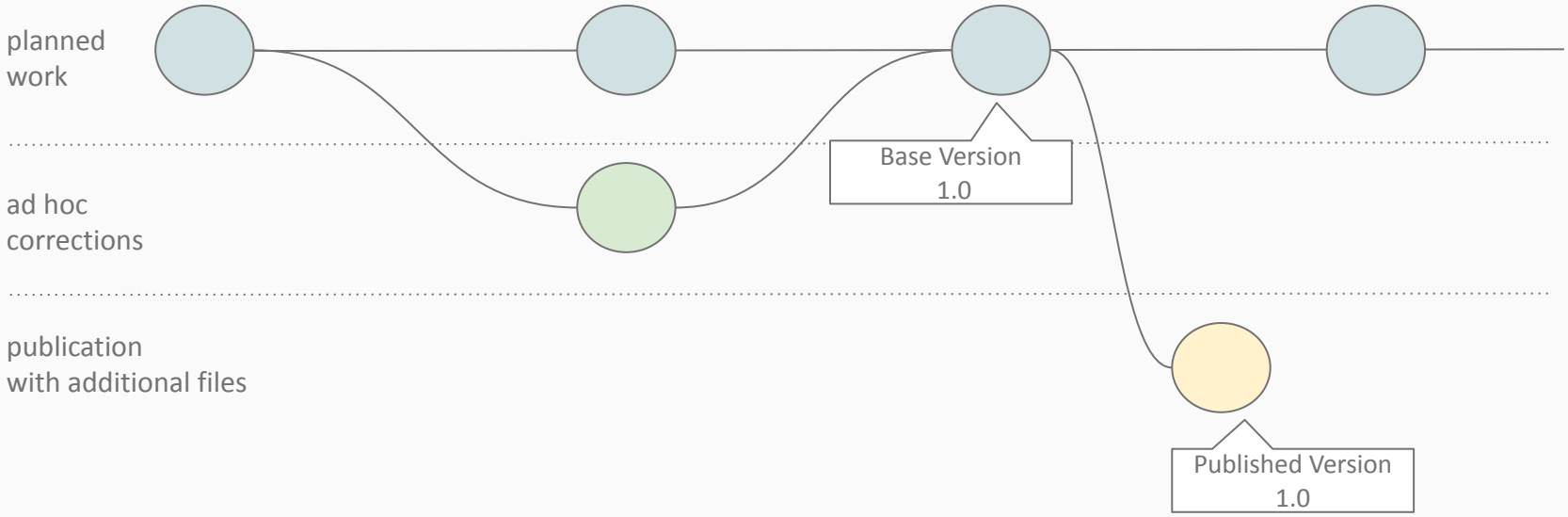
lokale Git-Installation:

- “git help log” bei Git Bash oder im Terminal eingeben

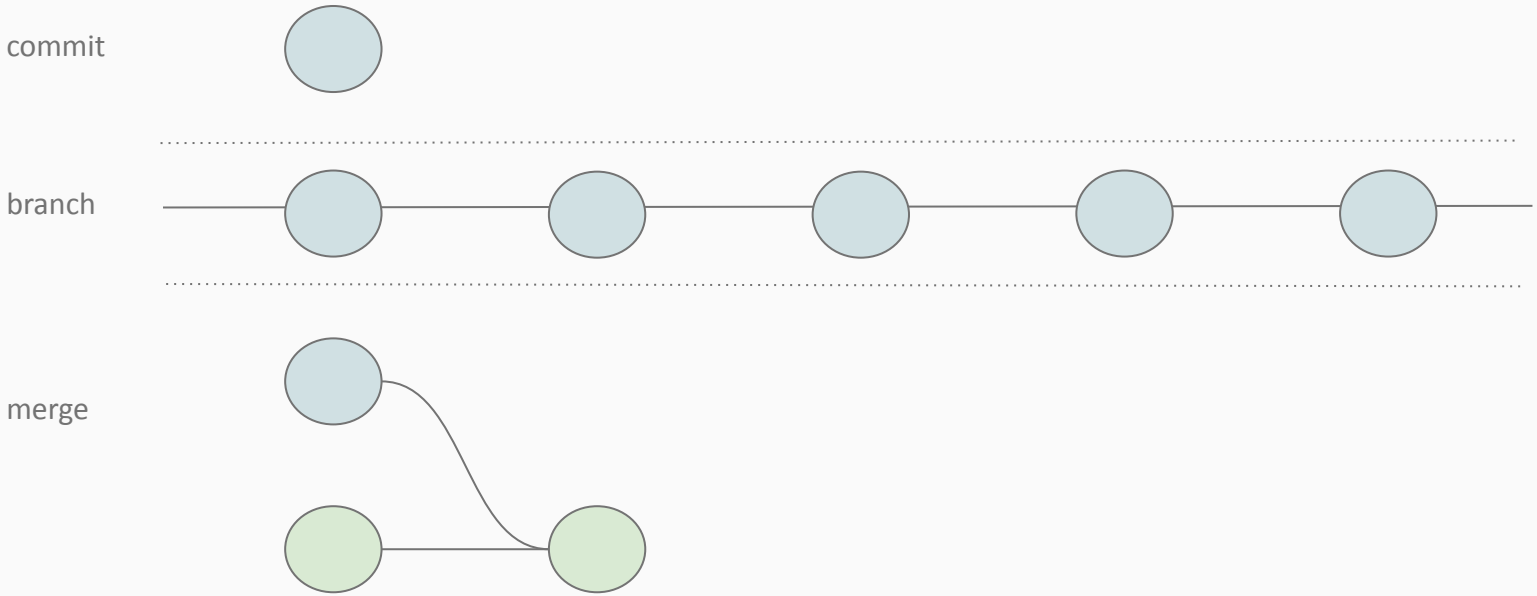
GitHub Desktop:

- Help > Show user guides

Git-Grundlagen



Git-Grundlagen



Git-Grundlagen

- Distributed VCS - was bedeutet das für uns?
  - GitHub und lokal auf unserem Computer können unterschiedliche Versionen desselben Repos liegen
  - bare repo vs. local repo

Git-Grundlagen

- Git: VCS (wie SVN)
- GitHub: Server für das VCS (wie GitLab, BitBucket)
- GitHub Desktop: Ein Programm, das die Nutzung von Git vereinfacht/visualisiert

(Forschungs-)Daten in Git

Welche Daten können oder sollten mit Git versioniert werden?

- Git wurde vor allem für Softwareentwicklung/Code entwickelt
- textbasierte Daten
- binäre Dateien
- XML Daten
- Besonderheiten bei Forschungsdaten

Erste Git-Konfiguration

$ git config --global user.name "Your Name Comes Here"

$ git config --global user.email you@yourdomain.example.com

Oder GitHub Desktop:

File>Options...>Git

Name und Email angeben/ändern

Andere Git-Projekte nutzen

- Lizenzen beachten
- Test-Beispiel: https://github.com/digitalhumanists/funwithvcs

  $ git clone https://github.com/digitalhumanists/funwithvcs

Oder GitHub Desktop:

File>Clone repository...>URL

URL zu Repository und lokalen Pfad angeben

Andere Git-Projekte nutzen

- Git-History
- branches, merges, pull requests
- forks

Eigenes Git(Hub)-Projekt erstellen und strukturieren

$ git init -b main

Oder GitHub Desktop:

File>New repository

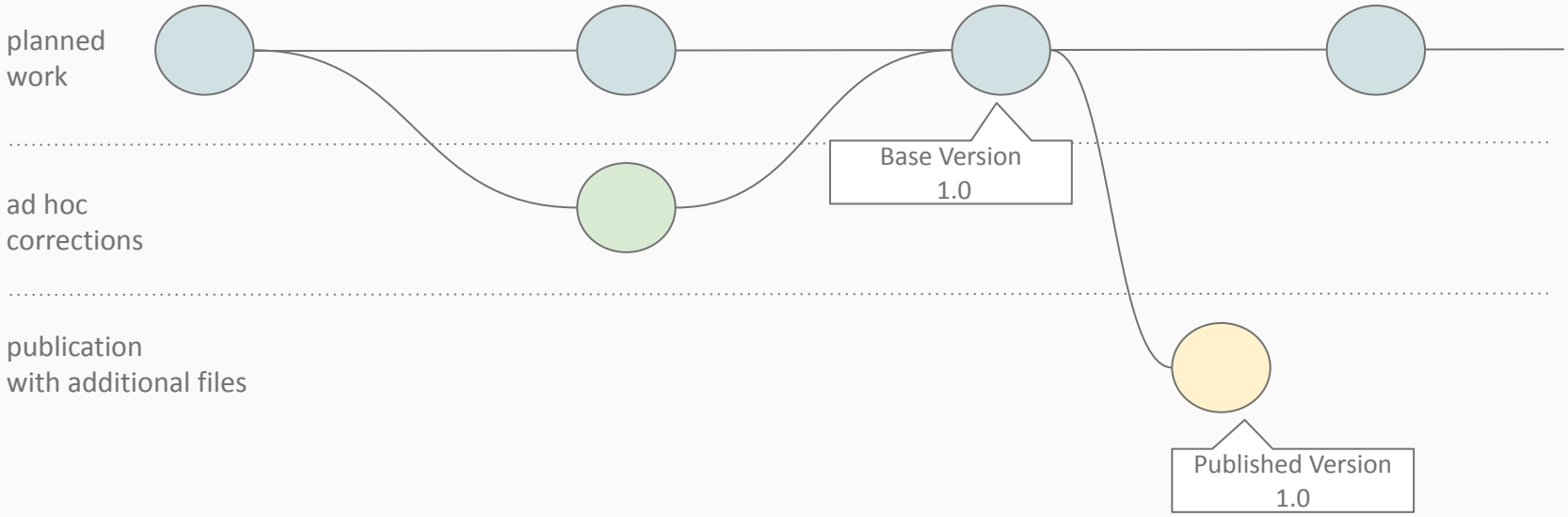
Eigenes Git(Hub)-Projekt erstellen und strukturieren

- Konfiguration Git korrekt?
- GitHub Projekt erstellen
- existierende oder neue Daten/Code?
- .gitignore Datei

Git flow

- ein bestimmter Workflow beim Arbeiten mit Git
- feature branches und development (dev) branches
- es gibt verschiedene Git Flow Abläufe, je nach Projekt/Team
- GitHub Flow

Git flow


planned
work
ad hoc
corrections
publication
with additional files
Base Version
1.0
Published Version
1.0

Merge-Konflikte

- mergen von Dateien funktioniert bei Git automatisch
- Merge-Konflikte treten auf, wenn Git nicht automatisch (text- und zeilenbasiert) erkennen kann, wie die unterschiedlichen Versionen zusammengeführt werden können
- dann muss manuell die korrekte Version aus den beiden vorherigen Versionen erstellt werden

Merge-Konflikte

Lösen von Merge-Konflikten:

- Konfliktdatei (das ist die ursprüngliche Textdatei, aber mit Konfliktmarkern) in einem Texteditor öffnen
- Konfliktmarker entfernen und nur gewünschte Version behalten

- Alternative für grafische Darstellung der Konflikte: Meld

Besonderheiten bei Forschungsdaten

- binäre Dateien
- Lizenzen und Datenschutz
- Veröffentlichung der Daten selbst

Verändern der History

- wenn gemeinsam an Repositories gearbeitet wird, sollte die History immer intakt gelassen werden, auch wenn fehlerhafte Commits in der History sind
- bei alleine genutzten Repos oder sehr problematischen Commits (Passwörter) kann die History aber trotzdem verändert werden
- Alle Commits bekommen dann aber eine neue SHA und es kann Probleme geben, wenn noch eine alte Version des Repos vorhanden ist

Weiterführende Möglichkeiten für Git und GitHub

- CI (continuous integration)
- Releases
- Webseiten mit github.io
- GitHub Actions
- Automatische Publikation mit Zenodo
- Automatische Zitation mit cff Dateien