Einführung in die Softwareentwicklung

SOMMERSEMESTER 2020

Institut für Informatik

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

```cpp
// This is C++
void main() {
    int var = 42;
    cout << "Hello World!";
}

# This is Python!
var = 42
print("Hello World!")
```

Vorlesung #11

Versionsverwaltung

Michael Wand · Institut für Informatik · michael.wand@uni-mainz.de

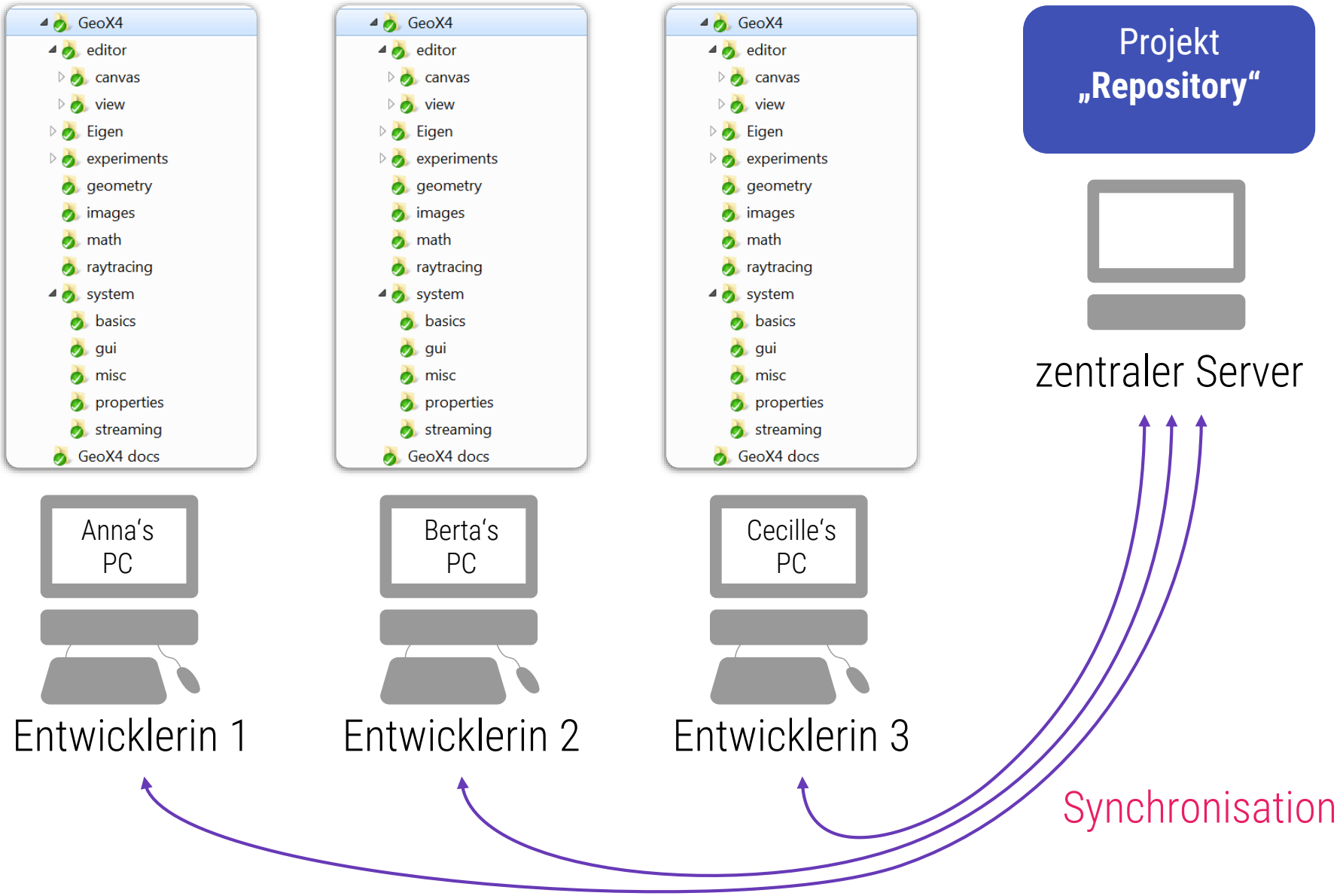
Versionsverwaltung

Versionsverwaltung

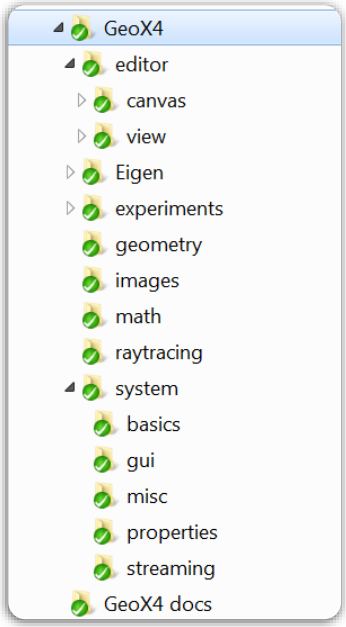
Ziele

- Alle Änderungen nachverfolgen
  - „Gestern tat es noch… :-(“
  - Alte Versionen können restauriert / verglichen werden
- Kollaboration
  - Mehrere Entwickler/innen
  - Synchronisation von verschiedenen Änderungen
  - „Konflikte“ (widersprüchliche Änderungen) werden erkannt
- Kommunikation
  - Zentraler Aufbewahrungsort („Cloud“)
  - Alternativ: dezentrale Synchronisation

Prinzip (Beispiel CVS, SVN)



Prinzip (Beispiel CVS, SVN)



Projekt
„Repository“

update

Update holt änderungen
Konflikte werden markiert!
Merging: Konflikte beheben

commit

Neue Version hochladen
Neuster Stand nötig (vorher update)
Änderungen fest (Transkation)
Änderungshistorie (immutable history)

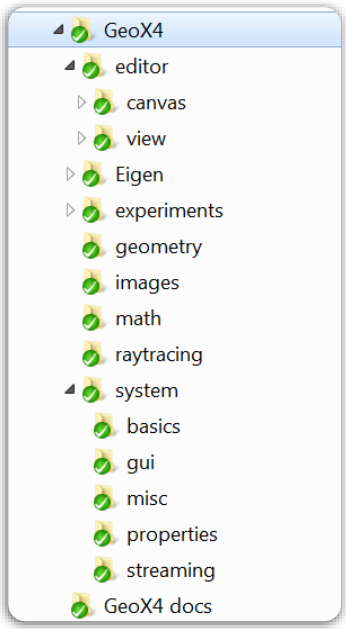
zentraler Server

Anna‘s
PC

Entwicklerin

Synchronisation

Subversion (SVN)



init, import

Repo. anlegen, initial füllen

Projekt **„Repository“**

update

commit

checkout

Code holen (bestimmte Version)

zentraler Server

add: Dateien für Versionierung markieren

revert: Änderungen verwerfen

rename, move, delete: Dateien ändern

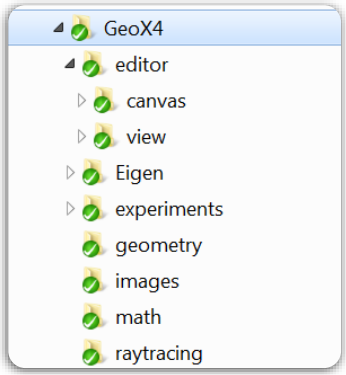
Anna‘s PC

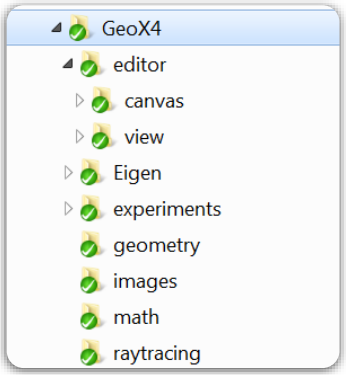
Entwicklerin

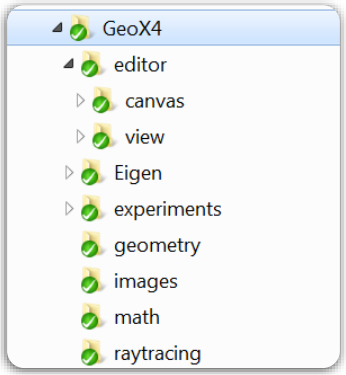
Synchronisation

Verteilte (dezentrale) Versionsverwaltung

Prinzip (Beispiel HG, GIT)



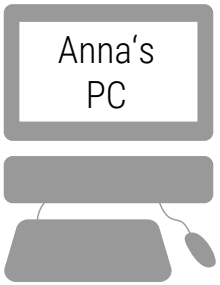


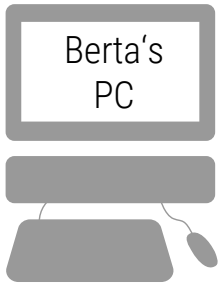


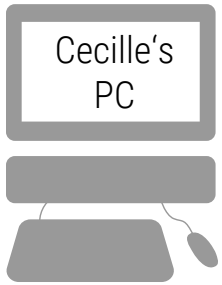
Anna's **Repository**

Berta's **Repository**

Cecille's **Repository**



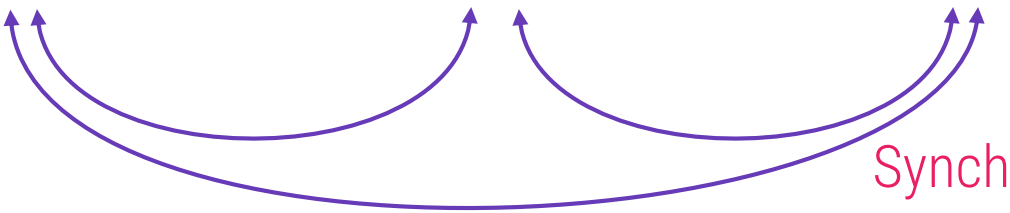




Entwicklerin 1

Entwicklerin 2

Entwicklerin 3

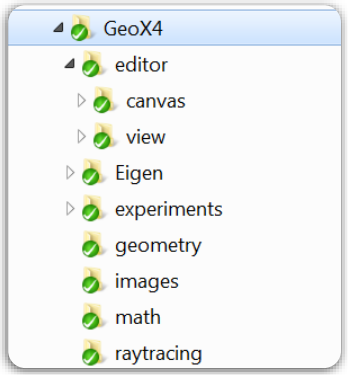


Synchronisation

kein
zentraler Server
nötig
(optional)

Mercurial (HG)

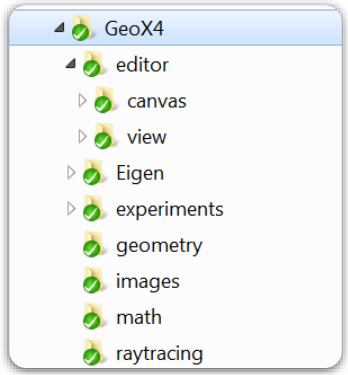
update, commit, add,move, delete
genau wie SVN



Anna's **Repository**

push
(Anna)

pull
(Anna)

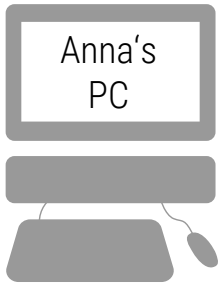


Berta's **Repository**

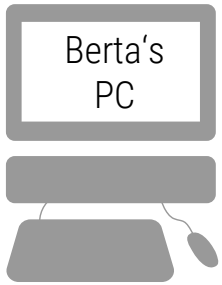
wie SVN

merge
nach pull
Branches
zusammen-
führen

Branching:
Verschiedene
Repoentwicklungen
werden unterschieden



Entwicklerin 1



Entwicklerin 2

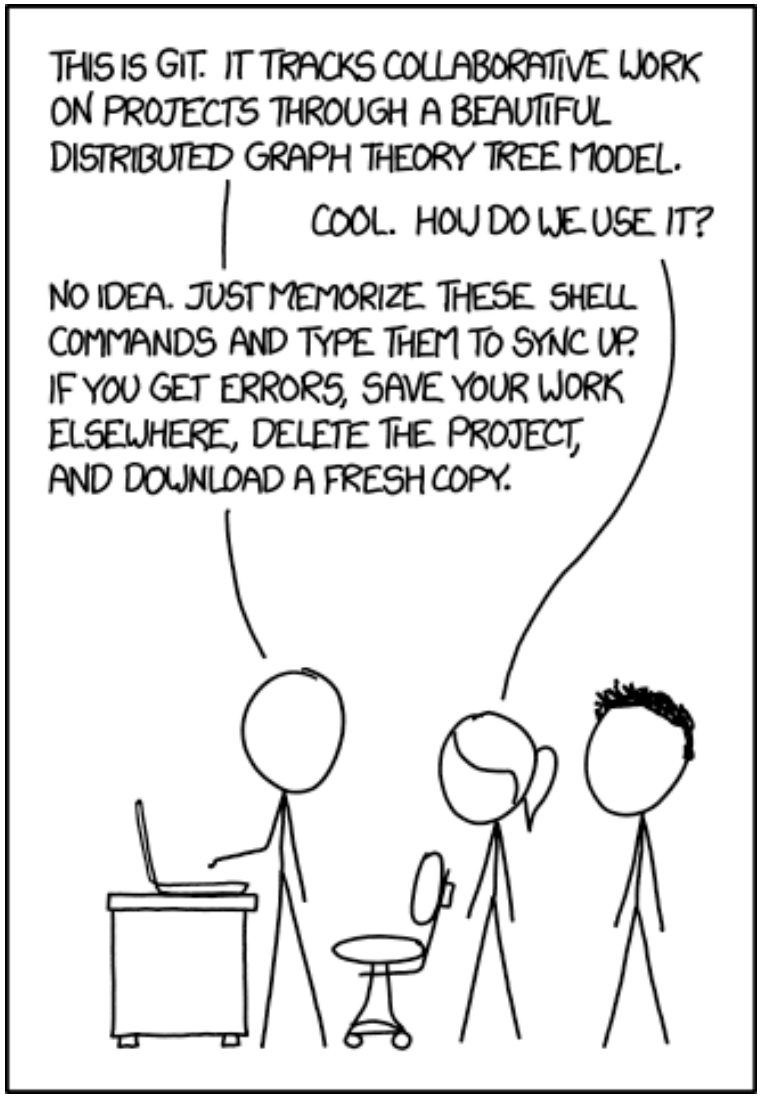
Zusätzlich:
Synchronisation der Repo's.

Was ist git?

In den Übungen

- Benutzung von „git“
- Gleiches Prinzip wie „hg“
  - Verteilte Versionsverwaltung
  - Mehr Features
  - Leider deutlich schwerer zu bedienen
  - „Industriestandard“
  - Hauptvorteil: „History rewriting“ möglich
    - Kann im kleinen destruktiv sein
    - Bei großen Projekten wichtig, um Ballast zu entfernen
- Ähnliche Situation wie bei C++ (komplex aber beliebt)

Mehr zu git in den Übungen...



https://xkcd.com/1597