

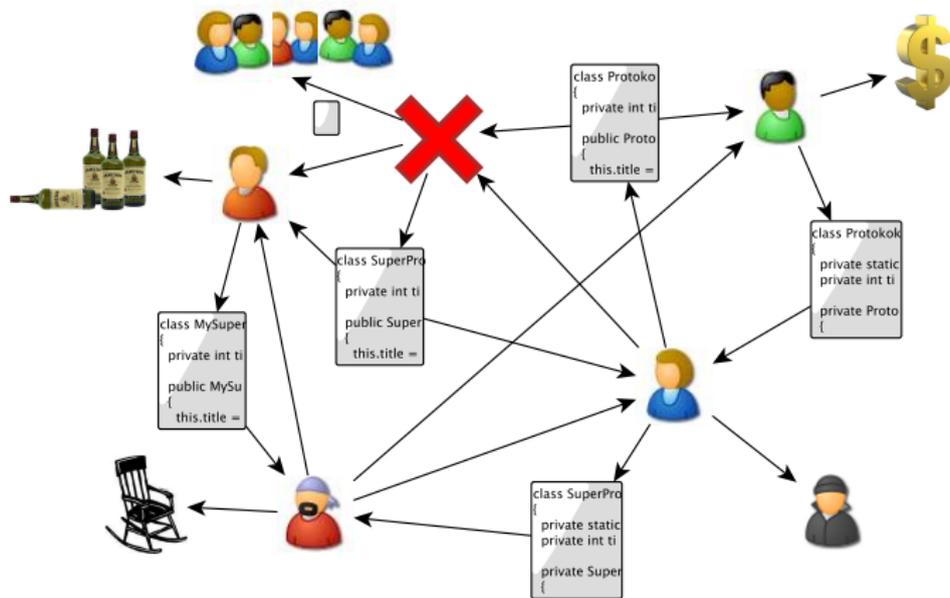
Einführung in Subversion (SVN)

Julian Fietkau

Universität Hamburg

28. August 2009

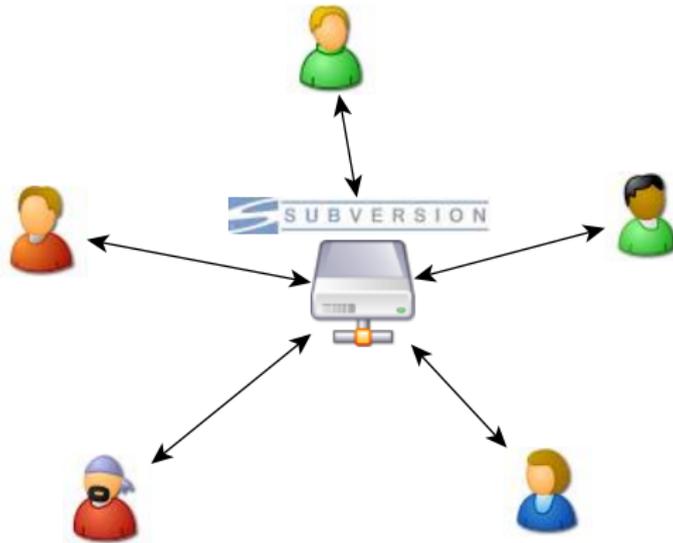
Programmieren im Team



Wie verhindert man ein solches Chaos?



Mit Subversion!





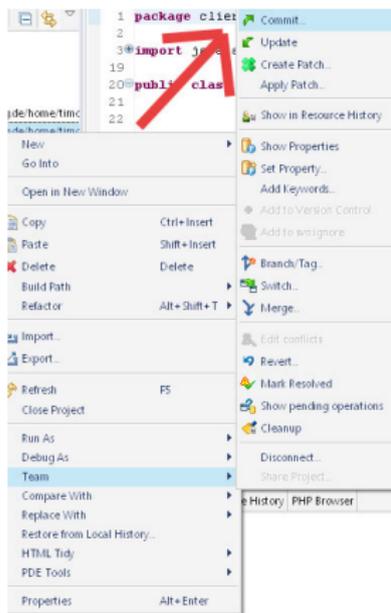
- Versionsverwaltungssystem
- **zentralisiertes** Versionsverwaltungssystem
- Projektdaten liegen auf einem zentralen Server im **Repository**
- Wenn jemand eine Datei bearbeiten will:
 - Datei **auschecken**
 - bearbeiten
 - und wieder **einchecken (commit)**
- Wenn zwei Leute das tun: assistierte Konfliktbehandlung

- Commits sind **atomar** (ganz oder gar nicht)
- Jede eingetragene Änderung erzeugt eine neue **Revision**
 - ...und erhöht dadurch die aktuelle **Revisionsnummer**
- Beim Commit kann (soll!) ein Kommentar (**Commit Message**) angegeben werden
- Mehrere parallele Entwicklungspfade (**branches**) sind möglich
- Bearbeitungskonflikt: Unterschiede (**diff**) werden dargestellt und können eingearbeitet werden

Man kann Subversion über die Kommandozeile bedienen...

```
user$ cd /home/user/svn
user$ svn update
user$ nano Datei.java
user$ svn commit -m "foo" Datei.java
```

Die Benutzung (2)



... es geht aber auch mit Eclipse: **Subclipse**

- Eclipse-Plugin mit Anbindung an SVN
- Rechtsklick in der Projektübersicht → im Menü „Team“ steht alles Wichtige
- Muss nachinstalliert werden

Commit early, commit often!
(sofern der Code kompiliert)



Wir wollen **code bombs** vermeiden.

- Subversion: <http://subversion.tigris.org/>
- Subclipse: <http://subclipse.tigris.org/>
- TortoiseSVN: <http://tortoisesvn.tigris.org/>
- Auf Wikipedia:
[http://de.wikipedia.org/wiki/Subversion_\(Software\)](http://de.wikipedia.org/wiki/Subversion_(Software))
- Zum Nachlesen: <http://svnbook.red-bean.com/>