

CVS

A. Kleiner
Universität Freiburg

SoPra 2003

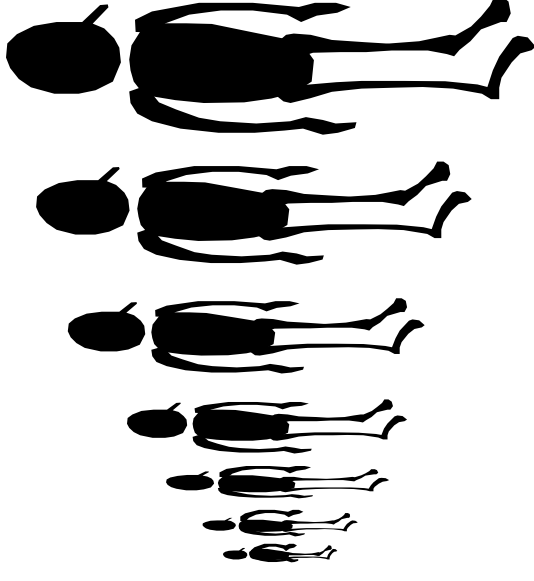
Problem: Mehrere Entwickler und eine Software

- Was kann man tun?
 - Aufteilung der Module?
 - Jeder arbeitet an einem eigenem Modul
 - Zusammenführen am Schluss
 - Warum geht's plötzlich nicht?
 - Alle auf einem Verzeichnis?
 - Hmm, irgendwie fehlen meine letzten Änderungen ...
 - Cannot open
 - Wie kommt man an Zwischenversionen ran?
 - Backups?
 - Jede Stunde?
 - Nach jeder Änderung?
 - Immer Sonntags?



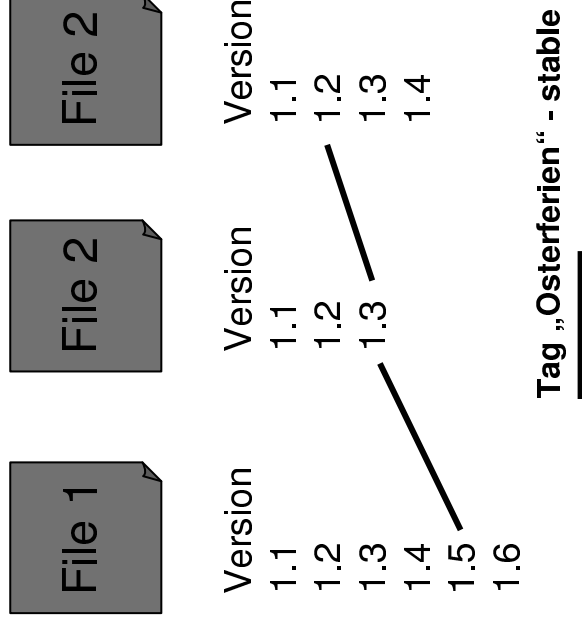
CVS (Concurrent Version System)

- Zentrale Code-Datenbank für mehrere Benutzer
 - Jeder Benutzer hat seinen eigenen Code
 - Änderungen werden durch „**commit**“ abgeschlossen
 - Holen der Änderungen durch ein „**update**“
 - Lock-Mechanismus, ähnlich wie bei Datenbanken
- Gleichzeitige Änderungen an einer Datei werden durch ein „**merge**“ bewerkstelligt
 - Hierbei stellen nur Änderungen an der gleichen Stelle in der Datei eine Schwierigkeit dar



CVS (Concurrent Version System)

- CVS speichert Änderungen, nicht Zustände
 - Jede Datei hat eine Versionsnummer die bei Änderungen erhöht wird
 - Alte Versionen können zurückgewonnen werden
- Tag Mechanismus:
 - Der komplette Zustand kann markiert und später abgerufen werden
- Log Mechanismus:
 - Dadurch fällt es Anderen leichter die Änderungen nachzuvollziehen
 - Sämtliche Transaktionen werden aufgezeichnet
 - Jedes „commit“ muss mit einem Kommentar versehen werden (z.B. *Habe Speicherloch entfernt*)



CVS

Die wichtigsten Befehle

- Einrichten:
 - Angabe des CVS Verzeichnisses unter Verwendung der Umgebungsvariable CVSROOT
 - `tcsh> setenv CVSROOT /home/gruppe01/repository`
 - `bash> export CVSROOT=/home/gruppe01/repository`
 - Liegt das Repository auf einem anderen Rechner, kann auch per `scp` darauf zugegriffen werden:
 - `>setenv CVS_RSH ssh`
 - `>setenv CVSROOT student@kaspar:/home/gruppe01/repository`
 - Zusätzlich sollte noch der bevorzugte Editor und der CVS Befehl angegeben werden:
 - `>setenv CVSEDITOR vi`
 - `>setenv CVS_SERVER /usr/bin/cvs`
 - Alle Einstellungen können dauerhaft in der Datei `$USERNAME/.login` gespeichert werden

CVS

Die wichtigsten Befehle

- Anlegen eines Repositories:
 - `>cvs init` (Bezieht sich immer auf CVSROOT)
- Importieren eines Projektes:
 - `>cd myAgents`
 - `cvs import -m "Imported sources" myAgents v_tag r_tag`
- „Auschecken“ eines Projektes
 - `>cvs checkout myAgents`
- „Updaten“ eines Projektes:
 - `>cd myAgents`
 - `>cvs -d update (-d: Auch neue Verzeichnisse holen)`
- Änderungen „einchecken“
 - `>cd myAgents`
 - `>cvs commit` (Dann einen kleinen Text bez. Der Änderungen eingeben)

CVS

Die wichtigsten Befehle

- Dateien/Verzeichnisse hinzufügen
 - >cd myAgents
 - >cvs add ambulance.java
 - >cvs commit
 - Dateien entfernen:
 - >cd myAgents
 - >rm ambulance.java
 - >cvs remove ambulance.java
 - >cvs commit
 - Den momentanen Zustand markieren:
 - >cvs tag ourFirstStableVersion
 - Logs betrachten:
 - >cd myAgents
 - >cvs log ambulance.java
 - Vergleich: Mein code/Repository
 - >cvs status | grep Needs | grep Modi
- ```
/* * $Log: fieldLines.h,v $
* Revision 1.7 2002/04/01 16:25:25 alex
* new repository after crash
*
* Revision 1.6 2002/03/27 20:05:50 thilo
* added new localization method for field with poles as borders
*
* Revision 1.5 2002/03/12 17:21:40 alex
* Extension for playing without borders
*
* Revision 1.4 2001/07/29 21:36:55 markus
* Created new repository with new directory structure.
*
* Revision 1.3 2000/06/21 14:44:03 robocup
* Final 1999 version.
* All modules work with Qt 2.0.
* New makefile structure.
* * Revision 1.2 1999/03/18 10:05:58 hatzack
* Added file description, log message display and copyright note. */
```