CVS
Concurrent Versions System
Steilkurs

Software-Praktikum im Grundstudium
WS 2004/2005

Dipl.-Inform. Michael Kirchhof
Dipl.-Inform. Bodo Kraft
Prof. Dr.-Ing. Manfred Nagl

Department of Computer Science III
Software Engineering
Ahornstr. 55
52074 Aachen, Germany

Gliederung

- CVS – Was ist das?
- CVS – Was ist es nicht?
- Das Prinzip
- Die Basics

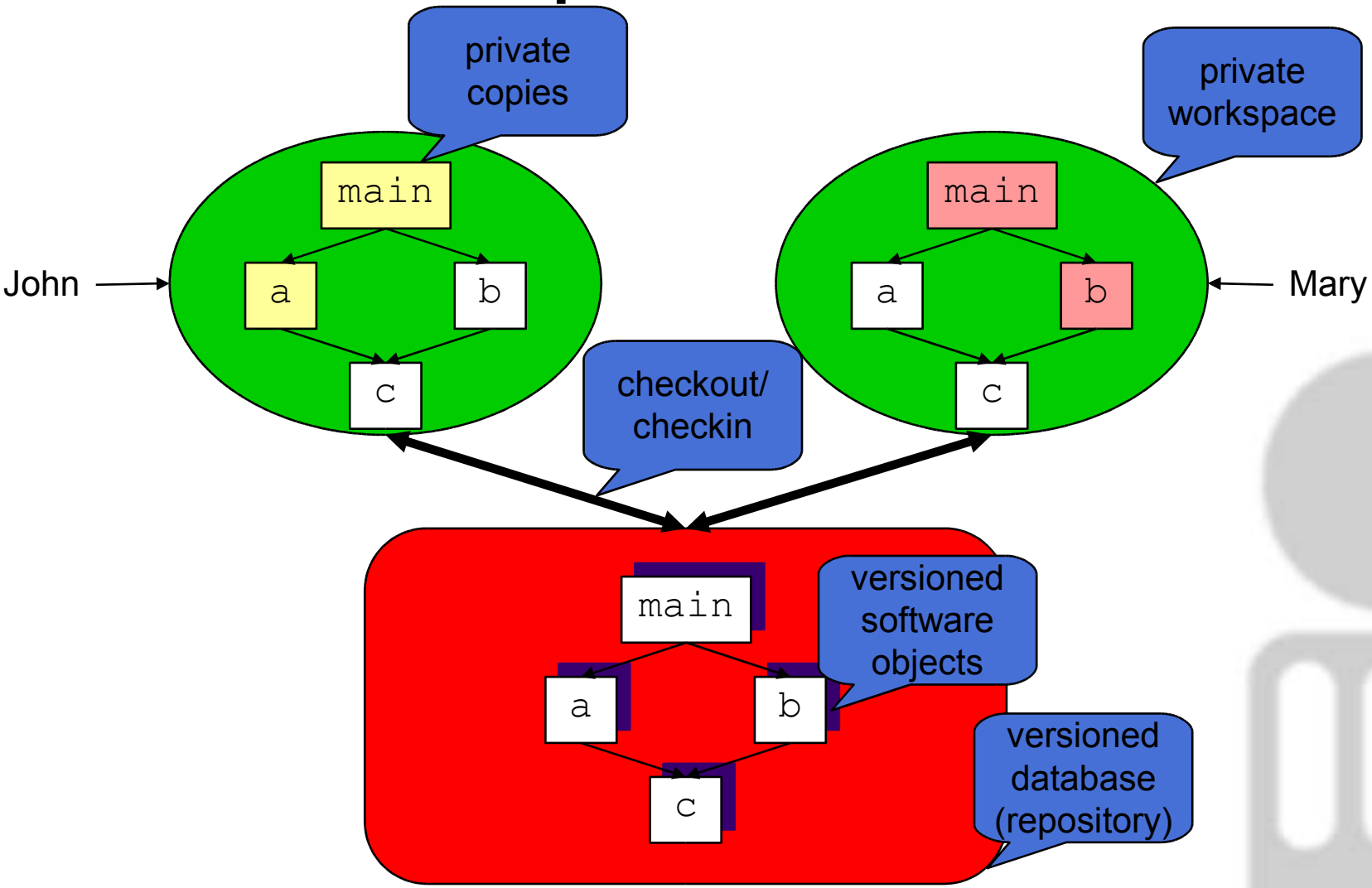
CVS: Was ist das?

- CVS is a version control system. Using it, you can record the history of your source files.
  - Typische Fragen (Babich 1986)
    - Gestern hat das funktioniert. Was ist seitdem passiert?
    - Bei mir tut’s das, aber warum?
    - Wo ist mein Bugfix von letzter Woche geblieben?
    - Warum sind meine Änderungen nicht in compilierten Programm?
    - Ist der Bugfix jetzt auch in meinen Dateien?
  - Bugs treten nicht direkt nach deren Entstehung auf und Weiterentwicklung verschleiern den Werdegang oft.
  - Warum speichert man dann nicht jede Version in einer neuen Datei?
- CVS also helps you if you are part of a group of people working on the same project. It is all too easy to overwrite each others' changes unless you are extremely careful.
- CVS solves this problem by insulating the different developers from each other. Every developer works in his own directory, and CVS merges the work when each developer is done.

CVS: Was ist es nicht?

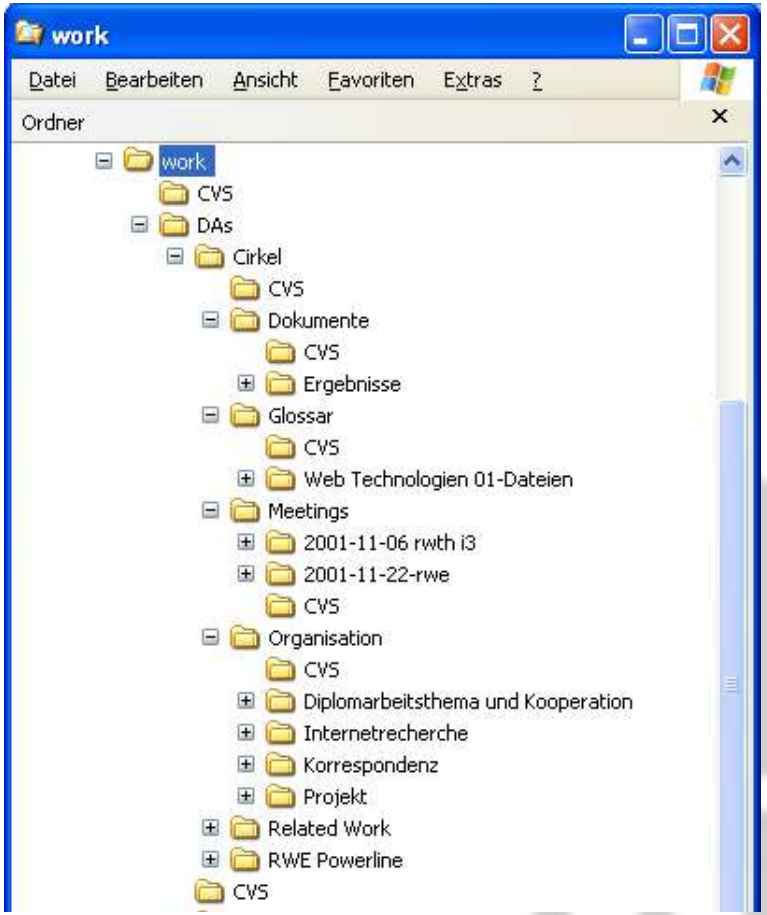
- CVS is not a build system.
- CVS is not a substitute for management.
  - CVS is an instrument for making sources dance to your tune. But you are the piper and the composer. No instrument plays itself or writes its own music.
- CVS is not a substitute for developer communication.
  - CVS cannot determine when simultaneous changes within a single file, or across a whole collection of files, will logically conflict with one another. Its concept of a "conflict" is purely textual, arising when two changes to the same base file are near enough to spook the merge (i.e. `diff3') command.
  - CVS does not claim to help at all in figuring out non-textual or distributed conflicts in program logic.
  - For example: Say you change the arguments to function `X' defined in file `A'. At the same time, someone edits file `B', adding new calls to function `X' using the old arguments. You are outside the realm of CVS's competence.
- CVS does not have change control
  - It can mean "bug-tracking", that is being able to keep a database of reported bugs and the status of each one (is it fixed? in which release? Has the bug submitter agreed that it is fixed?).
  - Keeping track of the fact that changes to several files were in fact changed together as one logical change. If you check in several files in a single `cvs commit' operation, CVS then forgets that those files were checked in together, and the fact that they have the same log message is the only thing tying them together.

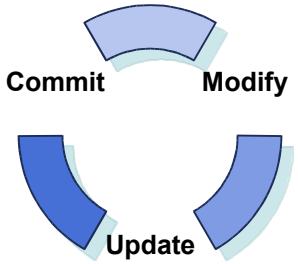
Das Prinzip


private copies
private workspace
main
main
John
a
b
a
b
Mary
c
c
checkout/ checkin
main
versioned software objects
a
b
versioned database (repository)
c

Die Basics

- Check-out
  - cvs checkout
- Check-in
  - cvs commit
- Diff
  - cvs diff
- Das CVS-Verzeichnis
  - Entries / Repository / Root
- CVSIGNORE
- Rekursives Verhalten
- Zugriffsprotokolle
  - Lokaler Zugriff
  - Pserver
  - ext



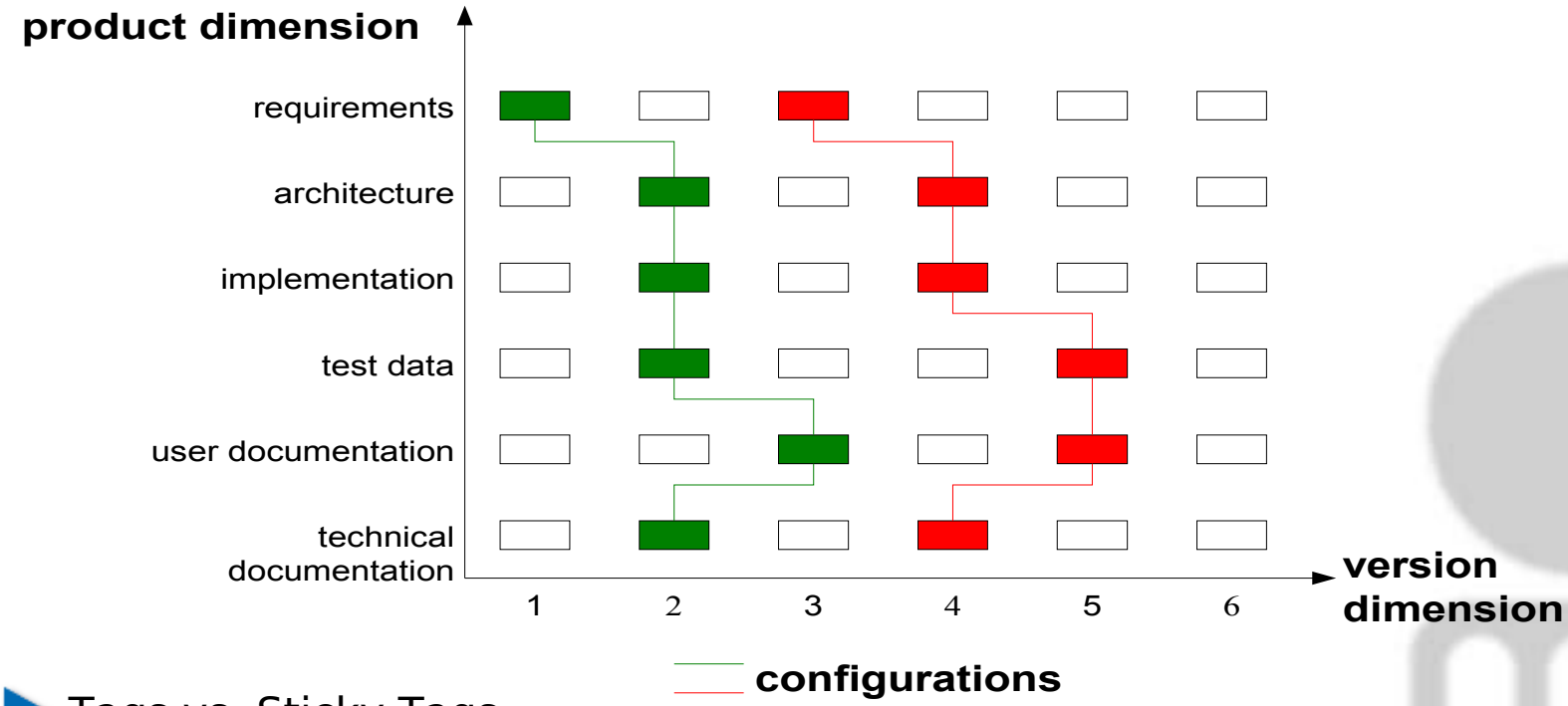


Bitte anschnallen...

- Tags
- Mehr-Benutzer-Betrieb
- Schlüsselwörter
- Binäre Dateien
- Administration

Bitte anschnallen... Tags (cvs tag)

- Warum?



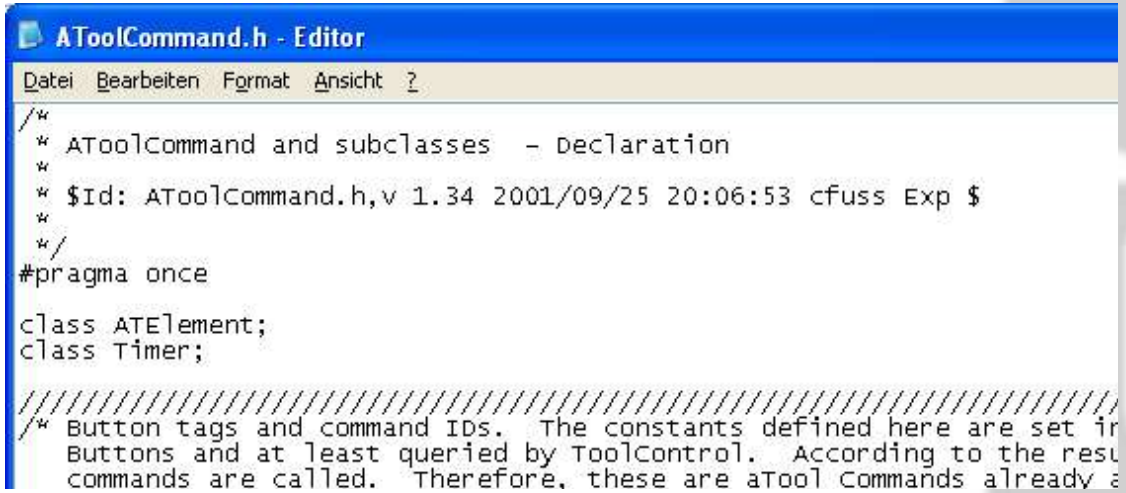
- Tags vs. Sticky Tags
- Tagging
  - cvs tag

Bitte anschnallen...Mehr-Benutzer

- Kommunikation
  - Commit-Messages
- Das berühmte Delta
- Konflikte

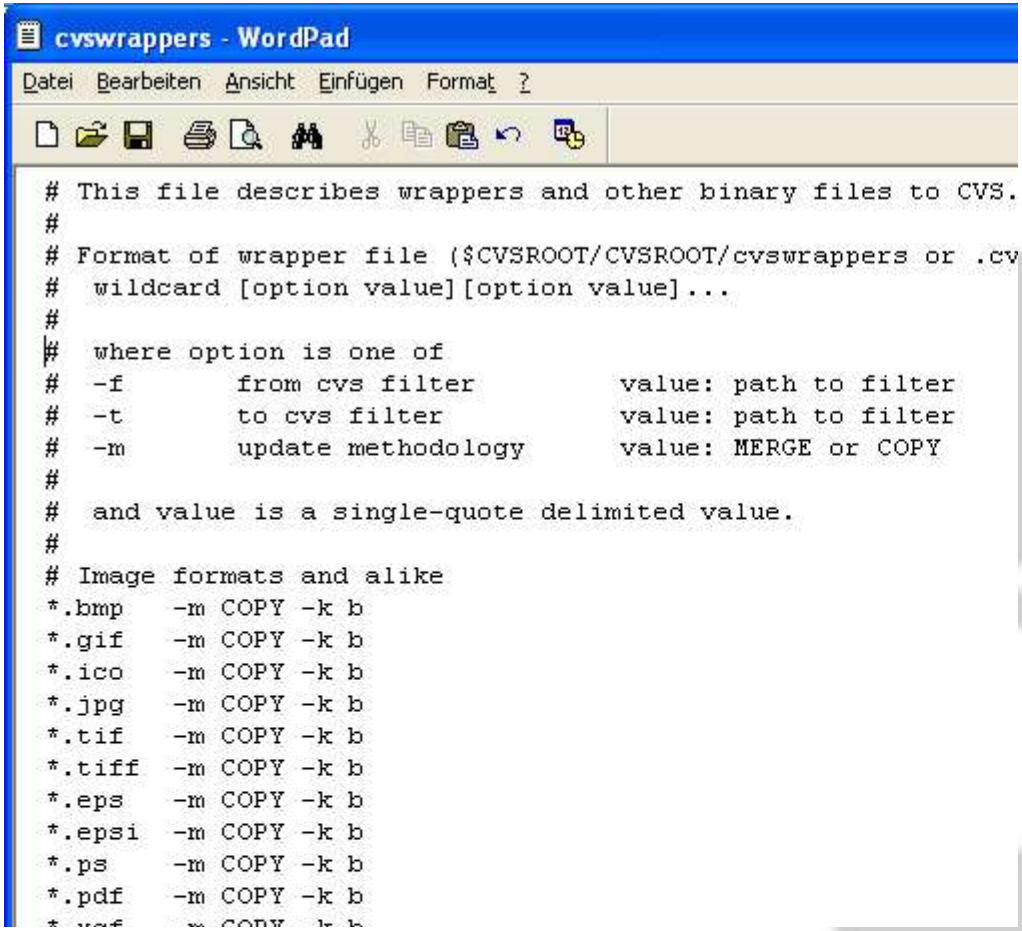
Bitte anschnallen...Schlüsselwörter

- Kennzeichnung/Annotation von verwalteten Dokumenten
- Verwendung
  - In nicht-binäre Dateien eingestreut
- Wichtige Schlüsselwörter
  - $Author$
  - $Date$
  - $Header$
  - Id
  - Log
  - $Revision$



Bitte anschnallen...Binäre Dateien

- Warum eine Sonderbehandlung?
  - Schlüsselwort-Expansion
  - Mergen vs. Copy
- Wie?
  - Cvswrappers



Bitte anschnallen...Administration

- Stets via Modul „CVSROOT“ in privater Arbeitskopie!
- checkoutlist
- cvsignore
- cvswrappers
  - *.bmp -m COPY -k b
- loginfo
  - DEFAULT mail -s "CVS Commit (default): %s" ppss02@i3...
- modules
- passwd
- writers
- readers

Weitere Quellen

- info cvs
- http://www.cvshome.org

- cvs –d ~spgs0405/Repository.Common checkout SampleDir
- cvs –d ~spgs0405/Repository.Common checkout CVSROOT

- Achtung:
  - CVS kennt keine Transaktionen!
  - CVS kennt kein Undo!

➔ Zum üben stets eine Sandbox verwenden!