

Versionsverwaltung mit Subversion

vom Verzeichnis zum Repository

Christian Dähn

Dipl.Inf. (FH)

Februar 2018

Agenda

- Motivation
- Grundlagen
- Arbeiten im Team
- Erweiterte Bedienung
- Literatur, weitere Infos

Motivation

- Source-Organisation ohne VCS
- Alltagsprobleme ohne VCS

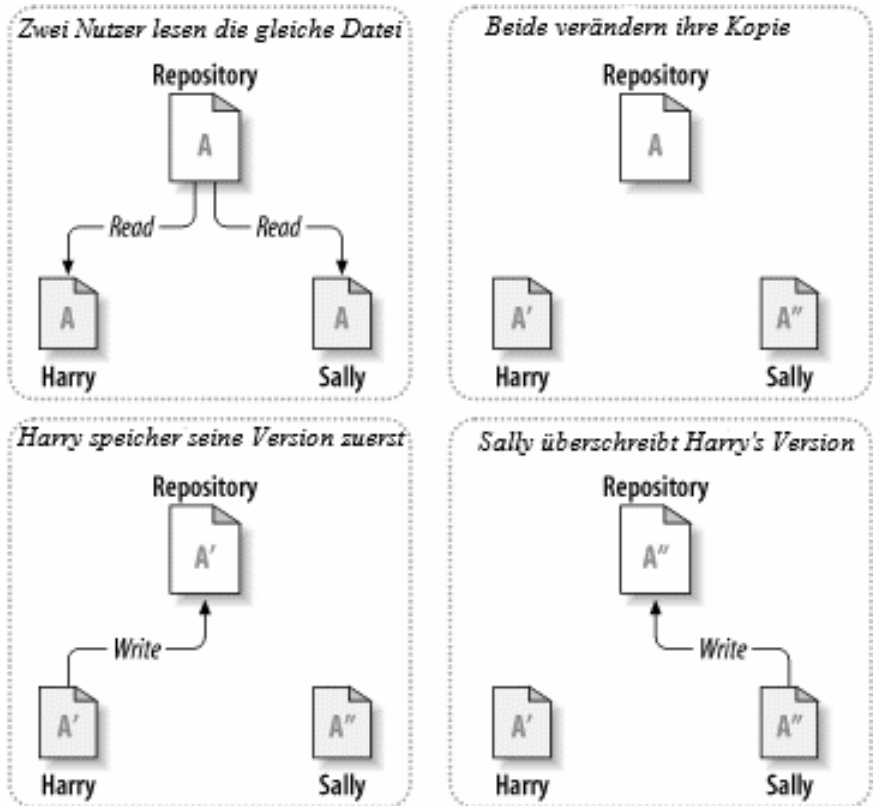
Motivation : Source-Organisation ohne VCS

Prinzip: Organisation in (Versions-)Verzeichnisse

- MyApp
 - 1.0.0
 - myapp.cpp
 - 1.1.0
 - myapp.cpp
 - setup.exe
 - setup.exe_new
 - 2.0.0
 - ...
 - old
 - ...

Motivation : Source-Organisation ohne VCS

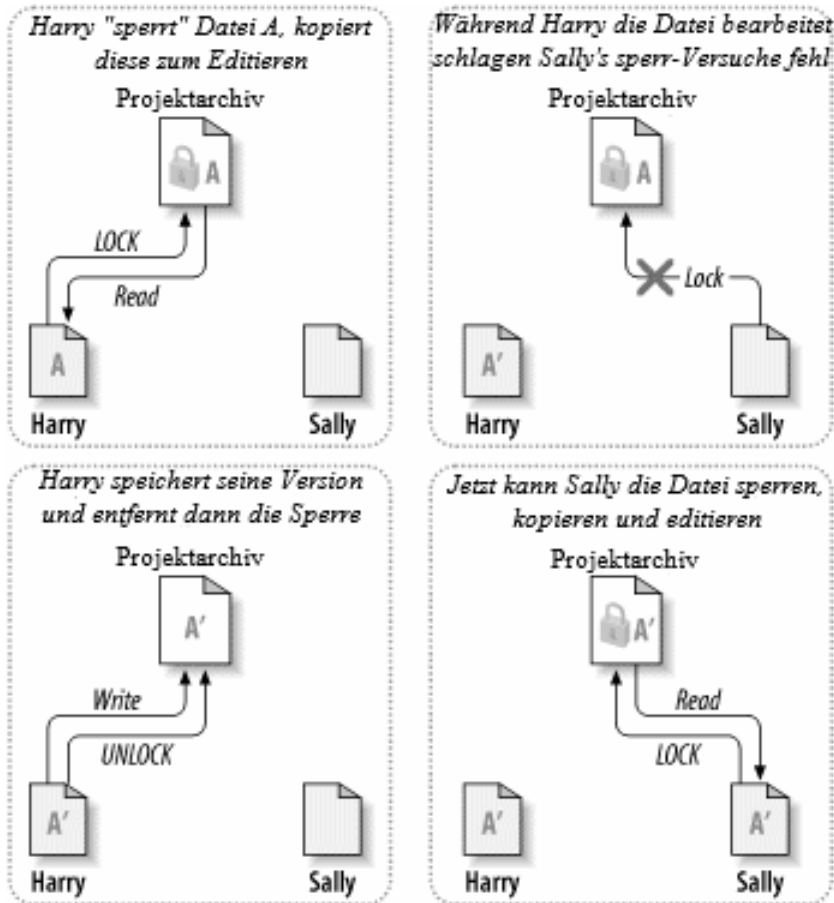
Problem: Gemeinsamer Dateizugriff



⚠ Manuelle Abstimmung + viel Kommunikation notwendig!

Motivation : Source-Organisation ohne VCS

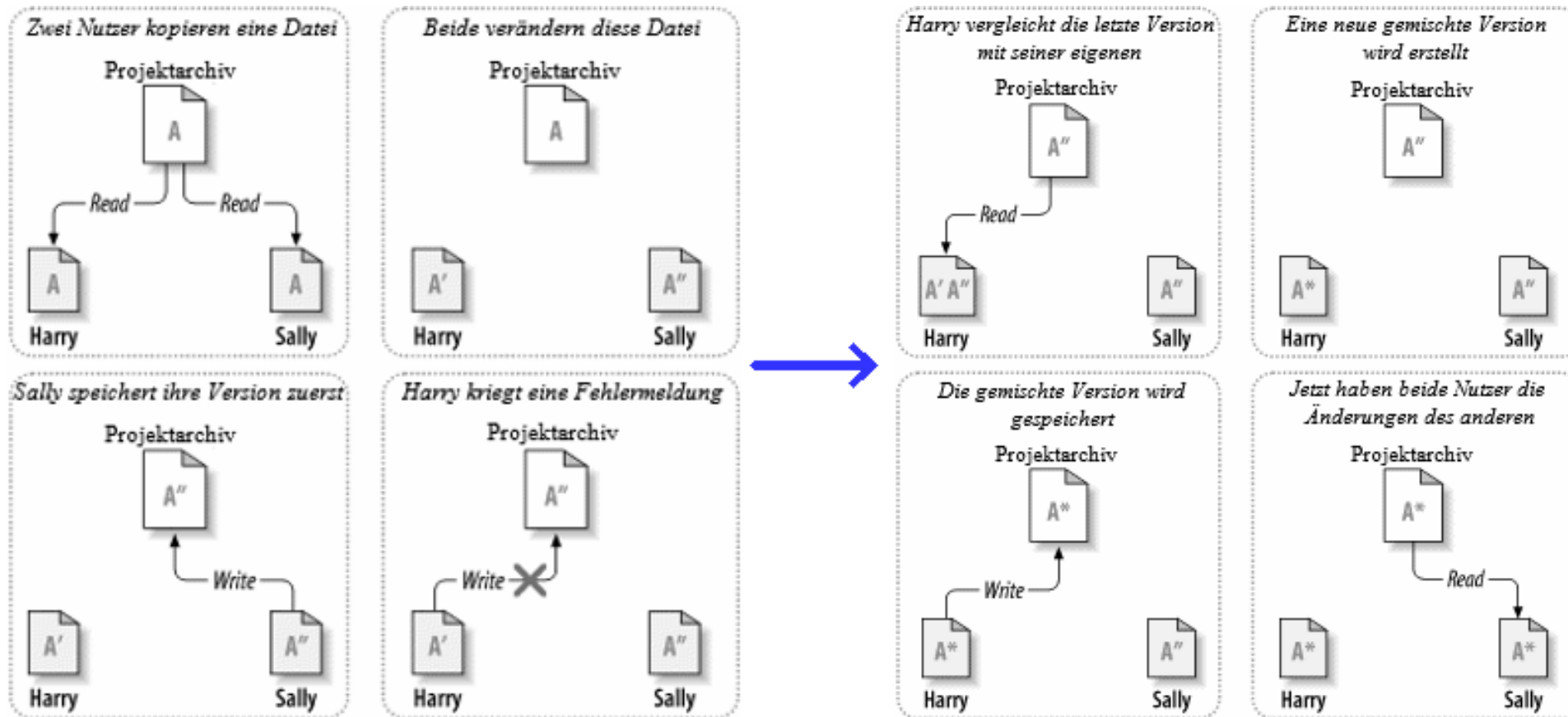
Workaround: Dateien sperren-ändern-freigeben



Keine parallele Bearbeitung, falsche Sicherheit...

Motivation : Source-Organisation ohne VCS

Lösung: Dateien kopieren-ändern-zusammenführen



⚠ Funktioniert nicht bei binären Dateien!

Motivation : Alltagsprobleme ohne VCS

Urlaub / Krankheit

- > *Code nicht verfügbar*
- > *Änderungen nicht für andere nachvollziehbar (Bug schon gefixed?)*

Störungen / Mc Murphy

- > *Datenverlust (Rechner defekt / abhanden gekommen, versehnl. Löschung)*
- > *Queltextstände versehentlich überschrieben*

Grundlagen

- Was sind VCS?
- Aufbau+Arbeitsweise von Subversion
- Clients
 - Apache Subversion
 - WebSVN
 - TortoiseSVN
- Übung

Grundlagen : VCS

Prinzip: Kopieren-ändern-Zusammenführen

--> *Nutzer arbeiten nur auf Kopien der Dateien*

Zentrale Datenhaltung auf Server = **Repository**

Integrierte Änderungsverfolgung

Integrierte Rechte- und Zugriffssteuerung

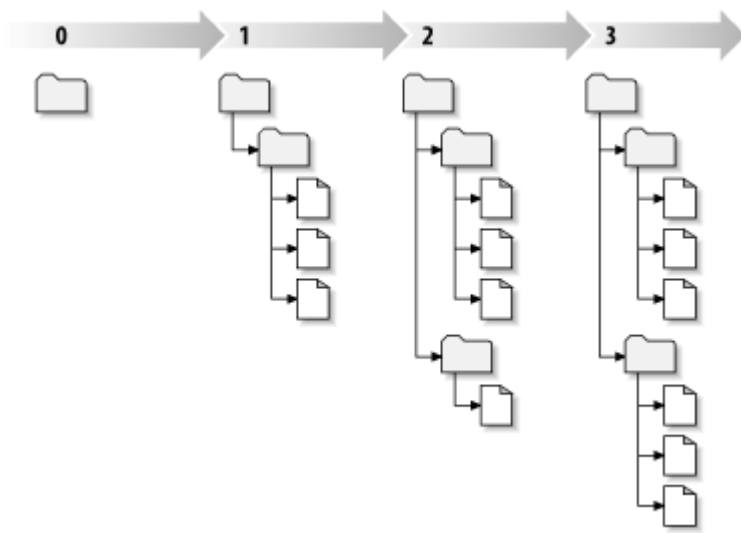
Grundlagen : Aufbau+Arbeitsweise von Subversion

Speicherung & Übertragung von Unterschieden, statt Dateien

--> Speicherung & Übertragung von Änderungen stets inkrementell

Repositories - physisch getrennte versionierte Verzeichnisse

--> globale Revisionen, inkrementiert bei jeder Änderung



Grundlagen : Aufbau+Arbeitsweise von Subversion

lokale "Arbeitskopien"

- > *Commits = Änderungen hochladen ändert keine lokalen Dateien!*
- > *Änderungen auf lokalen Daten müssen explizit angewiesen werden*

Konflikt-Erkennung und -Behandlung

- > *zeilenweise Vergleiche, Vorschläge für Konfliktbehandlung*

flache Kopien / copy-on-write

- *--> "copy" Befehl erzeugt Verweise, echte Kopien erst bei Änderungen

Grundlagen : Aufbau+Arbeitsweise von Subversion

Ordnerstruktur

```
/[SERVICE|PRODUKT]/[VARIANTE|PROJEKT]/[trunk|branches|tags]/
```

z.B.:

```
/SERiD/Bayern/trunk/main.cpp <-- Hauptentwicklungszweig
```

```
/SERiD/Bayern/trunk/doc/index.html
```

```
/SERiD/Bayern/trunk/tests/TestMain.cpp
```

```
/SERiD/Bayern/tags/Release 2.0/main.cpp <-- Kopie vom trunk
```

```
/SERiD/Bayern/branches/GUI-Redesign-1/main.cpp <-- Kopie vom trunk
```

Grundlagen : Clients

Gemeinsamkeiten

- Clone, Add, Commit, Update, Status ...
- Vergleiche, Änderungen verfolgen, Konflikte auflösen

Apache SVN (Command line)

Tortoise SVN (Windows)

- Explorer-Integration, eigene Diff- und Editor-Tools

Übung: FirstRepo auschecken und Datei hinzufügen

`http://kobudo.3dh.de:9280/svn/FirstRepo`

Arbeiten im Team

- Branching, Merging
 - Übung
- Bearbeiten einer Datei im Team
 - Übung

Erweiterte Bedienung

- Fehlersuche, Recherche
 - Übung
- Änderungen Wiederherstellen
 - Übung
- Releases verwalten
 - Übung

Literatur, weitere Infos

SVN Book ([kostenlos, HTML, deutsch](#))

TortoiseSVN ([Doku als HTML, Download](#))