

Kapitel 0.1 – Subversion (nicht prüfungsrelevant)

SWT I – Sommersemester 2021

Walter F. Tichy, Christopher Gerking, Tobias Hey

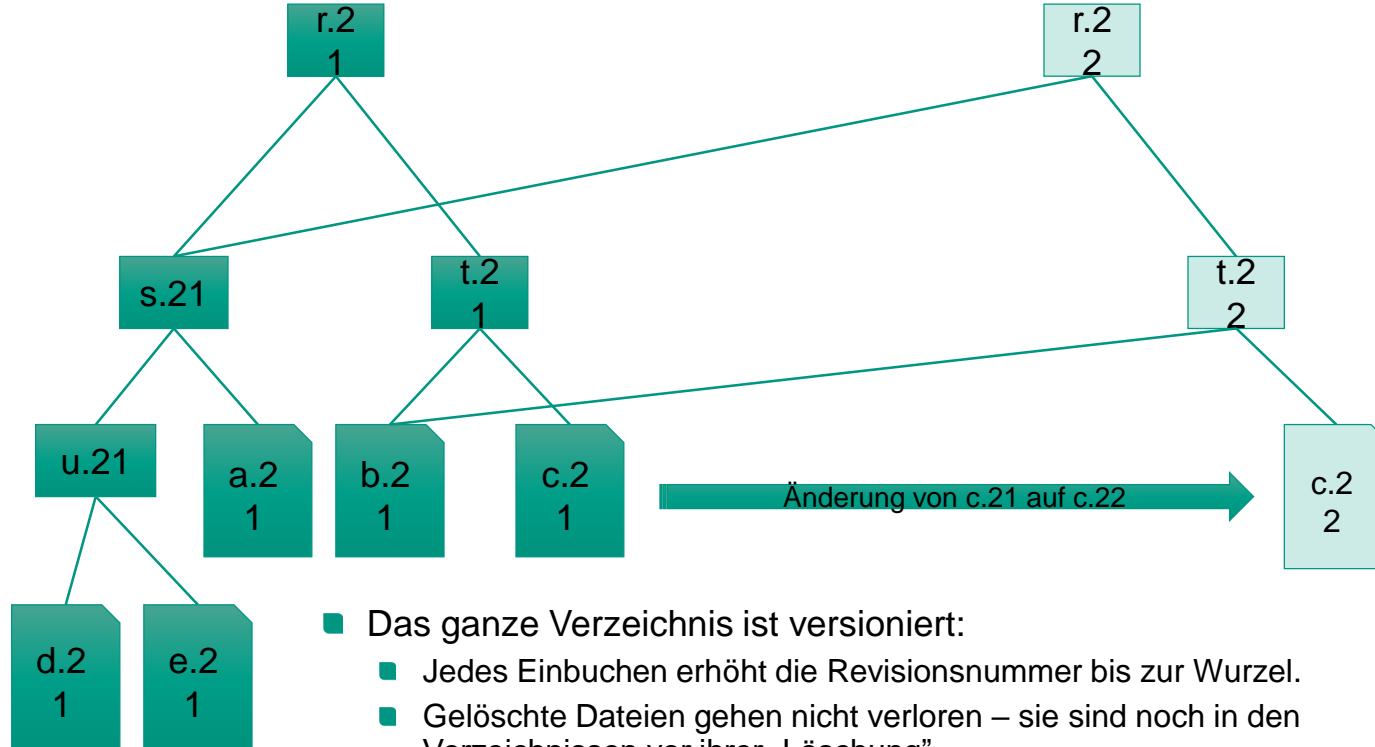


Konfigurationsmanagement mit **SUBVERSION**

Subversion (SVN, 2000)

- Weiterentwicklung von CVS
- Internetfähig, d.h. das Depot wird auf einem Dienstgeber abgelegt, auf den Arbeitsplatzrechner über das Internet zugreifen.
- SVN versioniert das gesamte Projektdepot, inklusive dem Verschieben, Umbenennen und Kopieren von Verzeichnissen und Dateien.
- Das Einbuchen von Änderungen geschieht atomar, d.h. eine Änderung wird ganz oder gar nicht in das Projektdepot übertragen.
- Optimistisches Ausbuchen

Versionierung von Verzeichnissen in SVN



- Das ganze Verzeichnis ist versioniert:
 - Jedes Einbuchen erhöht die Revisionsnummer bis zur Wurzel.
 - Gelöschte Dateien gehen nicht verloren – sie sind noch in den Verzeichnissen vor ihrer „Löschung“.

Varianten in SVN

- SVN bietet keine eingebaute Semantik für Variantenbildung.
- Statt dessen erzeugt man Unterverzeichnisse, die die Alternativen enthalten.
- Per Konvention benutzt man folgende Verzeichnisse:
 - **trunk**: Hauptentwicklungslinie
 - **branches**: alternative Entwicklungslinien, evtl. später mit Hauptentwicklungslinie verschmolzen.
 - **tags**: Kopien von anderen Verzeichnissen zu einem bestimmten Entwicklungsstand, benutzt um Freigaben einzufrieren.

Benutzerschnittstelle

- `svn <Kommando>`, z.B.
 - `help` – Liste der Kommandos
 - `help <Kommando>` – Detaillierte Hilfe zum Kommando
 - `import` – Einpflegen in Depot
 - Projekt in Depot erstellen
 - `checkout` – Verzeichnis/Projekt aus Depot laden
 - `commit` – Änderungen in das Depot übernehmen
 - `update` – Lokale Kopie aktualisieren
 - `add` – Neue Datei hinzufügen
 - `delete` – Datei aus Depot löschen
 - `move` – Datei/Verzeichnis verschieben
 - `copy` – Datei kopieren
 - `mkdir` – Verzeichnis anlegen

Anlegen eines SVN-Depots

- Liegt auf bestimmten Rechner im Netz, auf den alle zugreifen können.
- Der SVN-Server kann entweder eine eigenständige Installation sein oder in eine bestehende Apache-Webserver-Installation integriert werden.
- `svnadmin create`
 - Legt neues Depot auf dem Server an.
 - Beispiel:
`svnadmin create /var/svn/repos`

Verzeichnis/Datei einbuchen

■ `svn import [<Verzeichnis>] <URL>`

- Pflegt alle Dateien und Unterverzeichnisse des lokalen Verzeichnisses ein. Standardmäßig wird für `<Verzeichnis>` das aktuelle Verzeichnis „.“ angenommen.
- `<URL>` ist die Adresse zum Projektdepot, z.B. `https://.../repos`
- Mittels der Option `-m` kann eine zusätzliche Nachricht zur aktuellen Aktion angegeben werden.

Ausbuchen eines SVN-Depots

- Ausbuchen eines Depots bspw. mit dem Befehl:
`svn checkout https://.../repos /tmp/projects`
- Das Depot kann auch über weitere Protokolle wie bspw. `http://` etc. zur Verfügung gestellt werden.
- Benutzername und Passwort können mit Hilfe der Optionen `--username` und `--password` übergeben werden:

```
svn checkout https://.../repos /tmp/projects  
--username chuck --password sirron
```

Ausbuchen eines SVN-Depots

- `svn checkout <URL> [<Verzeichnis>]`
 - Legt lokal `<verzeichnis>` an
 - Kopiert alle Dateien des durch `<URL>` ausgewählten Verzeichnisses vom Depot nach `<verzeichnis>`
 - Legt Verzeichnis `<verzeichnis>/svn` mit Verwaltungsinformationen an
 - `.svn/` nicht löschen, nicht anfassen!
- Für den Rest der Kommandos müssen das Depot, Benutzername und Passwort nicht mehr angegeben werden. Es reicht z.B. `svn update`.

Lokale Dateien mit Depot synchronisieren

- `svn update <Verzeichnis/Dateien>`
 - Aktualisiert lokale Dateien mit neuen Versionen aus dem Depot, die seit Ausbuchen neu abgelegt wurden.
 - Mögliche Konflikte werden in separater Datei angezeigt (eigene Modifikationen werden nicht überschrieben!)
- Auflösung von Konflikten durch Verschmelzung muss man vor Einbuchen durchführen, da `svn commit` diesen Schritt nicht implizit durchführt!

Einbuchen I

- `svn commit <Verzeichnis/Dateien>`
 - Schreibt `<Verzeichnis/Dateien>` in Depot zurück
 - Bei Konflikten, bricht SVN ab → benutze `update` zuerst
 - Erhöht Revisionsnummer der veränderten Dateien.
 - Beim Einbuchen werden nur Änderungen der geänderten Dateien übertragen:
 - Im `.svn/` Verzeichnis befindet sich eine Kopie der letzten Version aller Dateien.
 - Zum Einbuchen werden die Änderungen zwischen diesen Kopien und den geänderten Dateien gesucht.
 - Diese Änderungen werden dann zum SVN-Server übertragen.
 - Neuer Verzeichnisbaum mit erhöhter Revisionsnummer wird angelegt, und alte und neue Revisionen von Dateien und Verzeichnissen eingehängt.

Einbuchen II

- Hat sich seit dem letzten Aktualisieren der lokalen Kopie die entsprechende Datei im Depot geändert, muss vor dem Einbuchen die **neue Version** im Depot mit der lokalen Kopie verschmolzen werden.
- Dazu bieten die graphischen Schnittstellen Werkzeugunterstützung an.

SVN Beispiel: Logbuch

Befehlseingabe: Ausführliches
Logbuch anfordern

```
tk@ipd:~/swt/proj/A/foo$ svn log -v bar
```

```
r2 | tk | 2011-04-11 22:35:19 +0200 (Mon, 11 Apr 2011) | 1 line
```

```
Changed paths:
```

```
  M /proj/A/foo/bar
```

Zweite Einbuchung:
Einbuchung einer
Änderung.

```
Test change.
```

```
r1 | tk | 2011-04-11 22:29:46 +0200 (Mon, 11 Apr 2011) | 1 line
```

```
Changed paths:
```

```
  A /proj
```

```
  A /proj/A
```

```
  A /proj/A/foo
```

```
  A /proj/A/foo/bar
```

```
  A /proj/B
```

Erste Einbuchung: Konfigurationsnummer,
Benutzername, Datum und Uhrzeit,
eingebuchte Dateien, Logbuchnachricht.

```
Initial commit.
```

SVN Beispiel: Delta

```
tk@ipd:~/swt/proj/A$ svn diff -r1:2 Hallo.java
Index: Hallo.java
```

```
-----
--- Hallo.java (revision 1)
```

```
+++ Hallo.java (revision 2)
```

```
@@ -4,7 +4,7 @@
```

```
}
```

```
private static String getHello() {
```

```
-     return "Hallo Uni!";
```

```
+     return "Hallo KIT!";
```

```
}
```

```
private static void doNothing() { }
```

Befehlseingabe:
Fordere das Delta
zwischen
Konfiguration 1 und 2
der Datei Hallo.java
an.

Angabe der Zeilen mit
Änderungen: In beiden Dateien
zwischen den Zeilen
4 bis 7.

Lies: „Um von Konfiguration
1 zu Konfiguration 2 zu
gelangen, entferne die Zeile
mit ‚Hallo Uni!‘ und füge die
Zeile mit ‚Hallo KIT!‘ ein.

SVN: Details

- Administrative Informationen befinden sich lokal im Unterverzeichnis `.svn/` (ähnlich wie bei CVS in `cv$`)
- Von jeder Datei befindet sich eine unveränderte Version im `.svn`-Verzeichnis. Dadurch können einige Befehle (wie bspw. `revert`) lokal ausgeführt werden.
- SVN unterstützt binäre Dateien. Es wird beim Einpflegen von Dateien automatisch erkannt, ob es sich um eine Text- oder Binärdatei handelt.
- Zur Zeichenkodierung verwendet SVN UTF-8, wodurch internationale Zeichen unterstützt werden.