

Verteilte Versionskontrolle mit SVK



Einführung und Nutzung
von SVK

FrOSCon 24./25. Juni 2006

Agenda

- Installation
- Grundbegriffe von SVK und Subversion
- Arbeit mit SVK
- „Problem“ von Subversion
- Online, Offline Arbeit
- Quellen

Voraussetzungen Unix

- Es muss Subversion mit den Perl Bindings installiert sein.

```
./configure -enable-swig-bindings=perl
```

- Konsequenz ist oft, dass Subversion neu kompiliert werden muss.

Voraussetzungen Windows

- Perl Installation
 - z.B. ActiveState
- Subversion Installation
 - Im PATH aufgenommen.
- Perl Bindings müssen installiert sein (svn-win32-1.X.Y_pl.zip)

Installation Linux

- Download SVK

```
tar -zxf SVK-1.06.tar.gz
```

```
perl Makefile.PL
```

```
==> Auto-install the XX mandatory  
module(s) from CPAN? [y]
```

```
make
```

```
make install
```

Installation Windows

- Download von SVK
 - Die Installation via `svk-1.06-svn-1.2.3.exe` ausführen.

Hinweis

- Es sollte überall die gleiche Version von Subversion installiert sein.

Erster Aufruf

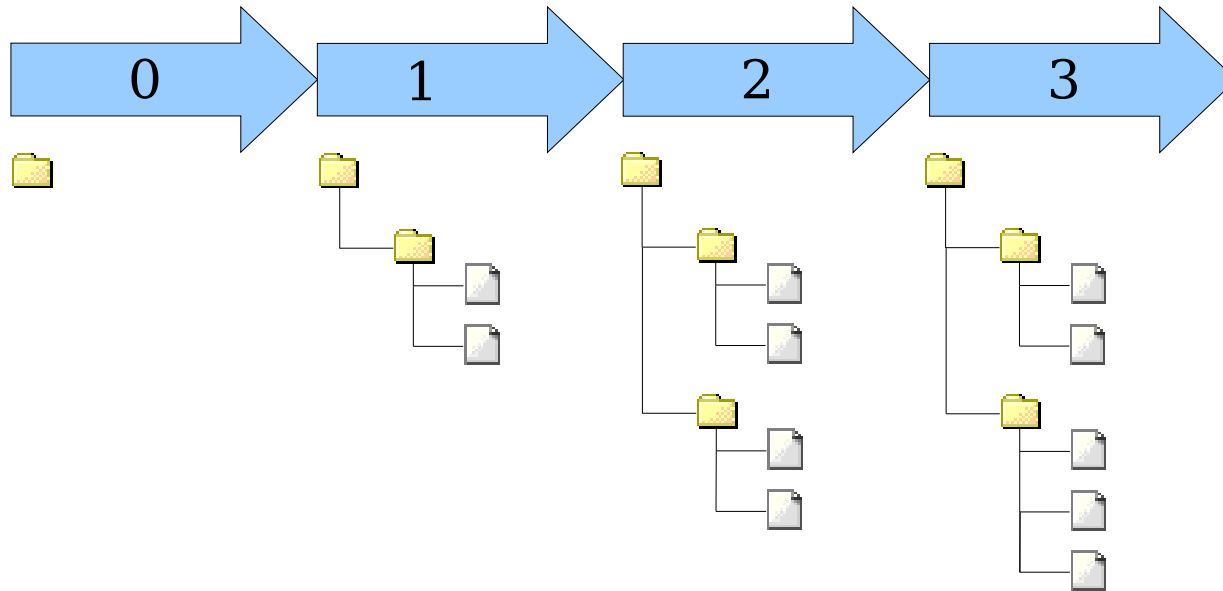
- Die Version von SVK:

```
svk --version
```


Konzept von Subversion

- Transaktionen gesichert.
- Dateien und Verzeichnisse werden als eine Einheit betrachtet.
- Eine einzige Revision Nummer für das gesamte Repository.

Revisions in Subversion



Begriffe

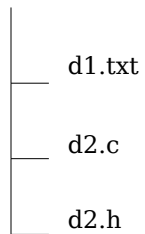
- Repository
- Einchecken/Auschecken
- Arbeitskopie (working copy)
-

Begriff: checkout

Arbeitskopie



projekt



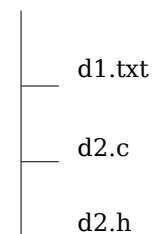
checkout



Repository



projekt



Begriff: checkin

Arbeitskopie

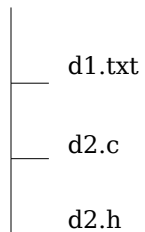


checkin

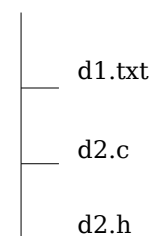
Repository



projekt



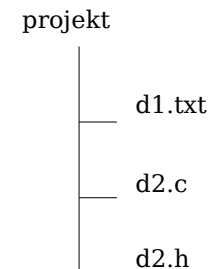
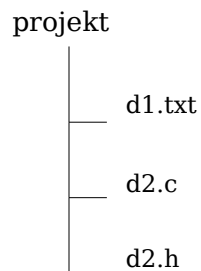
projekt



Begriff: Arbeitskopie

Arbeitskopie oder Working Copy (WC)

Repository



Hinweis

- Eine Arbeitskopie muss nicht zwangsläufig auf einem anderen Rechner liegen, wie das Subversion Repository. Sie kann auch auf dem gleichen Rechner liegen.

Es geht los!

- Einrichtung des Depots:

```
svk depotmap --init
```

```
Repository ~/.svk/local does  
not exist, create? (y/n)
```

- Hier mit “y” antworten.

Das Erste Projekt

- Import des ersten Projektes

```
svk import -m 'Log Message'
```

```
  // ProjectName/trunk
```

```
  /path/to/project
```

Das Erste Projekt

Committed revision 1.

Import path // **ProjectName**/trunk initialized.

Committed revision 2.

Directory E:\testsvk\exp imported to depotpath
// **ProjectName**/trunk as revision 2.

Hinweis:

- Die Struktur innerhalb eines Projektes mit den Verzeichnissen trunk, tags und branches sollte hier eingehalten werden.

Das Log-Kommando

- Log Anzeige

```
svk log //ProjectName
```

Das Log-Kommando

r2: kama | 2006-06-16 18:01:40 +0200

- PROJEKT Import into SVK

r1: kama | 2006-06-16 18:01:40 +0200

Directory for svk import.

Das Log-Kommando

- Log Anzeige

```
svk log -v //ProjectName
```

```
.....
```

```
- PROJEKT Import into SVK
```

```
-----  
r1: kama | 2006-06-16 18:01:40 +0200
```

```
Changed paths:
```

```
  A /ProjectName
```

```
  A /ProjectName/trunk
```

Projekt bearbeiten

- Wenn jetzt mit einem Projekt gearbeitet werden soll, muss es zuerst ausgecheckt werden.

Das Checkout-Kommando

```
svk checkout
```

```
  // ProjectName /trunk
```

```
  ArbeitskopieName
```


Besonderheiten von SVK

- In einer Arbeitskopie (working copy) sind keine speziellen Verzeichnisse oder Dateien zu finden wie beispielsweise bei Subversion oder CVS.

Das Checkin-Kommando

- In der Arbeitskopie:

```
svk checkin -m 'Log Message '
```

- Ohne die Option “-m” wird der Editor aufgerufen.

Arbeitskopie löschen?

- Entfernung einer Arbeitskopie
`svk checkout --detach`
ArbeitskopieName
- Danach kann das Verzeichnis
ArbeitskopieName gelöscht werden.

Arbeitskopien

- Wo liegen Arbeitskopien?

```
svk checkout --list
```

- Das ergibt eine Liste der Verzeichnisse, in denen Arbeitskopien liegen.

Änderungen feststellen

- Wie kann man feststellen, ob etwas geändert wurde?

```
svk status
```

```
? AllTests.java
```

```
M File1.txt
```

```
A File2.doc
```

Hinzufügen von Dateien

- Hinzufügen von Dateien/Verzeichnissen mithilfe von:

```
svk add FileDirectory
```

- Bei Verzeichnissen wird rekursiv hinzugefügt.

Umbenennen von Dateien / Verzeichnissen

- Umbenennen von Dateien /
Verzeichnissen

svk move ***Alt Neu***

Löschen von Dateien / Verzeichnissen

- Löschen von Dateien /
Verzeichnissen mithilfe von

`svk delete FileDirectory`

- Bei Verzeichnissen wird rekursiv
gelöscht.

Fehler gemacht! Und jetzt?

- Die meisten Änderungen, die in einer Arbeitskopie gemacht werden, können mithilfe von `svk revert File`
- wieder rückgängig gemacht werden.

Änderungen feststellen

- Wie kann man feststellen, **was** geändert wurde?

svk diff

Projektübersicht

- Wie kann man feststellen, **welche** Projekte im Depot vorhanden sind?

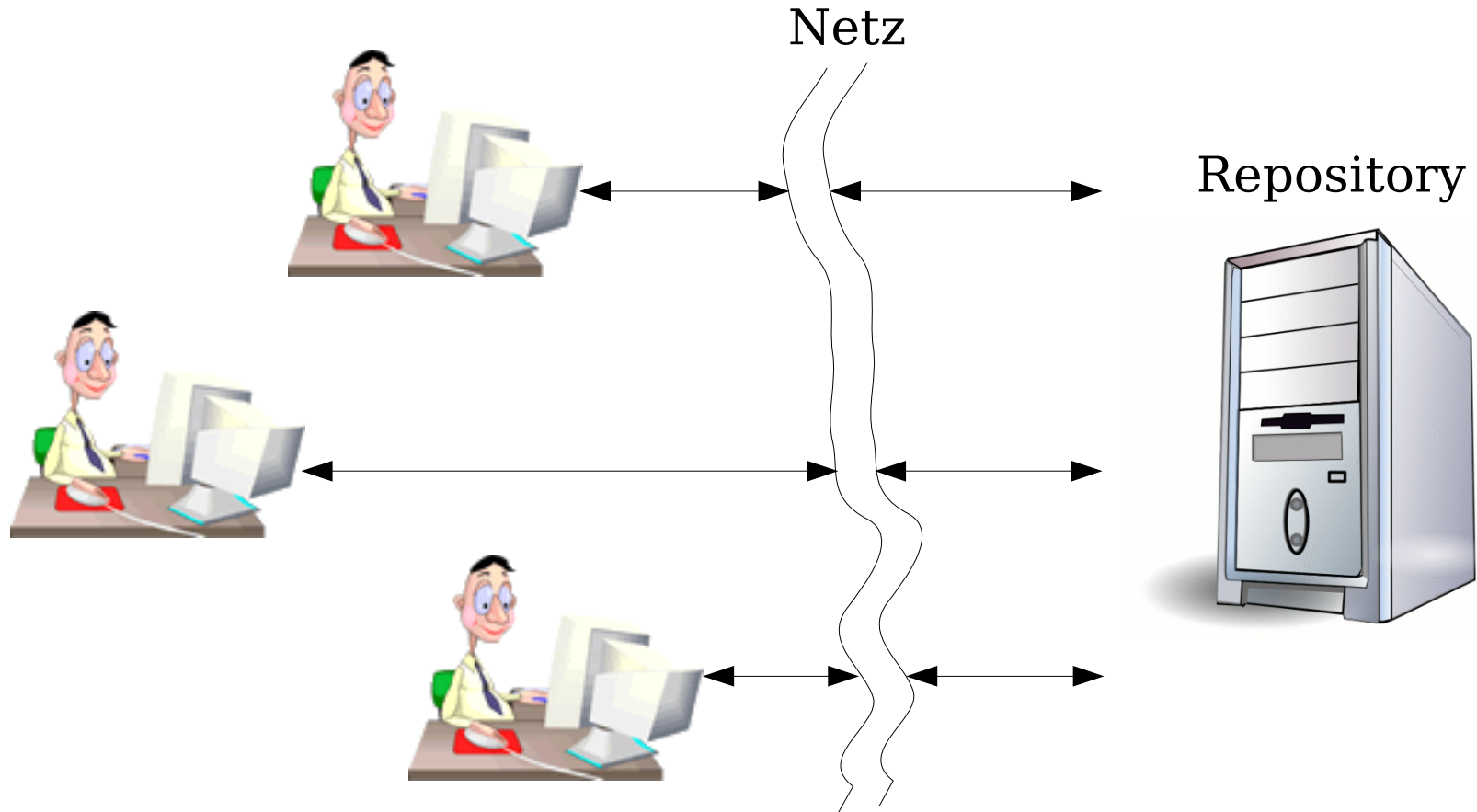
```
svk list //
```

Arbeitskopien?

- Wie kann man feststellen, **wo** Arbeitskopien ausgecheckt sind?

```
svk checkout --list
```

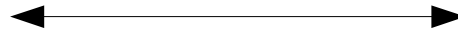
Konzept von Subversion



“Problem” von Subversion



?



Repository



“Problem” von Subversion

- Zum Ein-, Auschecken, Branching und Tagging (etc.) wird eine Netzwerkverbindung zum Repository benötigt.
- Ausnahme:
 - Lokale Repositories (file:///)

Auf Reisen

- Mithilfe von SVK kann das Problem gelöst werden.
- Wir müssen ein s.g. Mirror anlegen.

Mirror Kommando

- Mirroring eines Projektes

```
svk mirror
```

```
http://svktest.wanderer/ProjectName/trunk
```

```
//ProjectName/trunk
```

Angleichung

- Angleich des lokalen Projektes mit dem Remote Projekt.

```
svk sync //ProjectName/trunk
```

- Das kann je nach Größe einige Zeit in Anspruch nehmen.

Online Arbeit

- Auschecken.

```
svk checkout
```

```
// ProjectName/trunk
```

```
ProjectName
```

Online Arbeit

- Bearbeiten und einchecken.

```
svk checkin -m"- Log Message"
```

```
Commit into mirrored path: merging back directly.
```

```
Merging back to mirror source
```

```
http://svktest.wanderer/antsvk/trunk.
```

```
Merge back committed as revision 2.
```

```
Syncing http://svktest.wanderer/antsvk/trunk
```

```
Retrieving log information from 2 to 2
```

```
Committed revision 3 from revision 2.
```

Online Arbeit

- Hiermit wird jede Operation direkt auch auf das gespiegelte Repository ausgeführt.

Offline Arbeit

- Offline Arbeit benötigt einen lokalen Branch.

Offline Arbeit

- Nach dem `svk sync` muss noch Folgendes gemacht werden:

```
svk copy
```

```
// ProjectName/trunk
```

```
// ProjectName/local
```

```
-m" - Offline working"
```

Offline Arbeit

- Auschecken.

```
svk checkout
```

```
// ProjectName / local
```

```
ProjectName
```

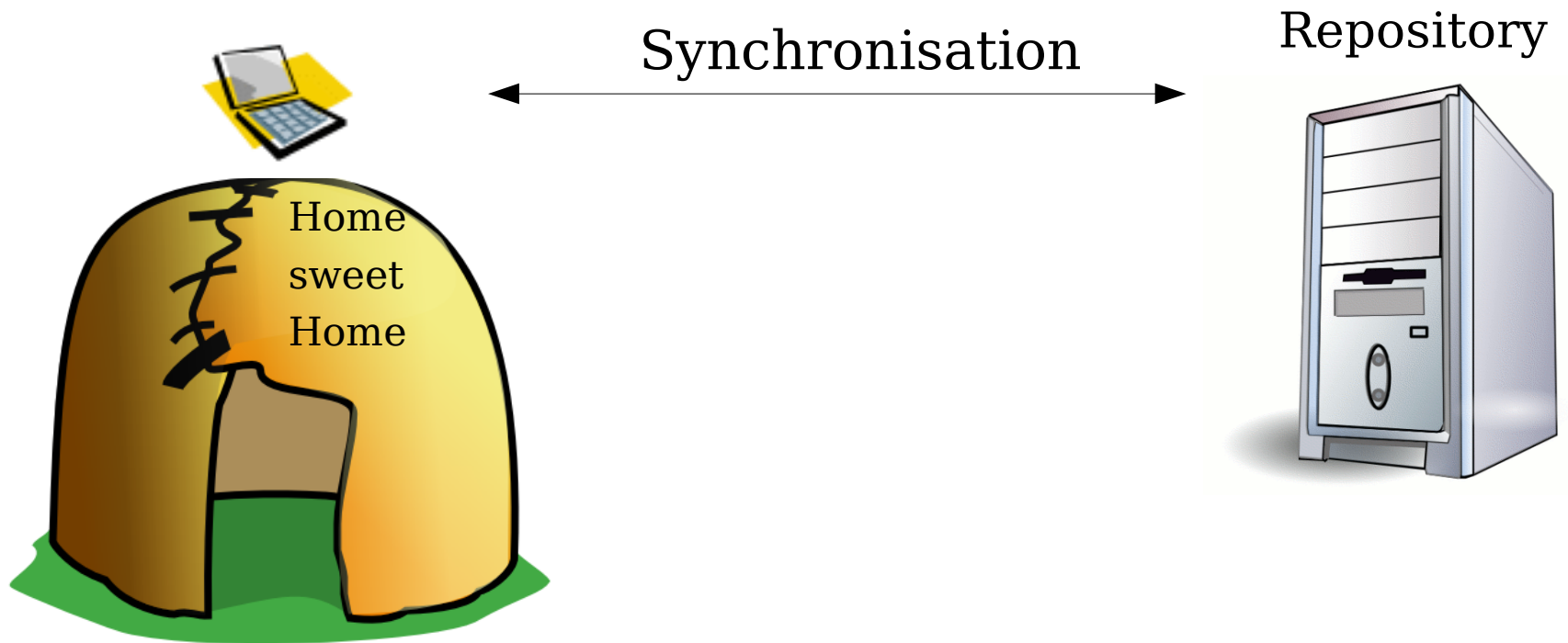

Offline Arbeit

- Bearbeiten und einchecken.

```
svk checkin -m"- Log Message"
```

```
Committed revision 5.
```

Synchronisation



Synchronisation

- Synchronisation zwischen dem lokalen Projekt und dem remote Projekt.

```
svk smerge -I -l  
//antsvk/local  
//antsvk/trunk.
```

Synchronisation

- Durch “-I” wird jede Änderung wie ein “checkin” auf dem Remote Projekt durchgeführt.
- Mit „-l“, wird die Log Message in das Remote Projekt übernommen.

Remote Änderungen

- Was passiert, wenn am Remote Projekt zwischenzeitlich Änderungen vorgenommen worden sind?

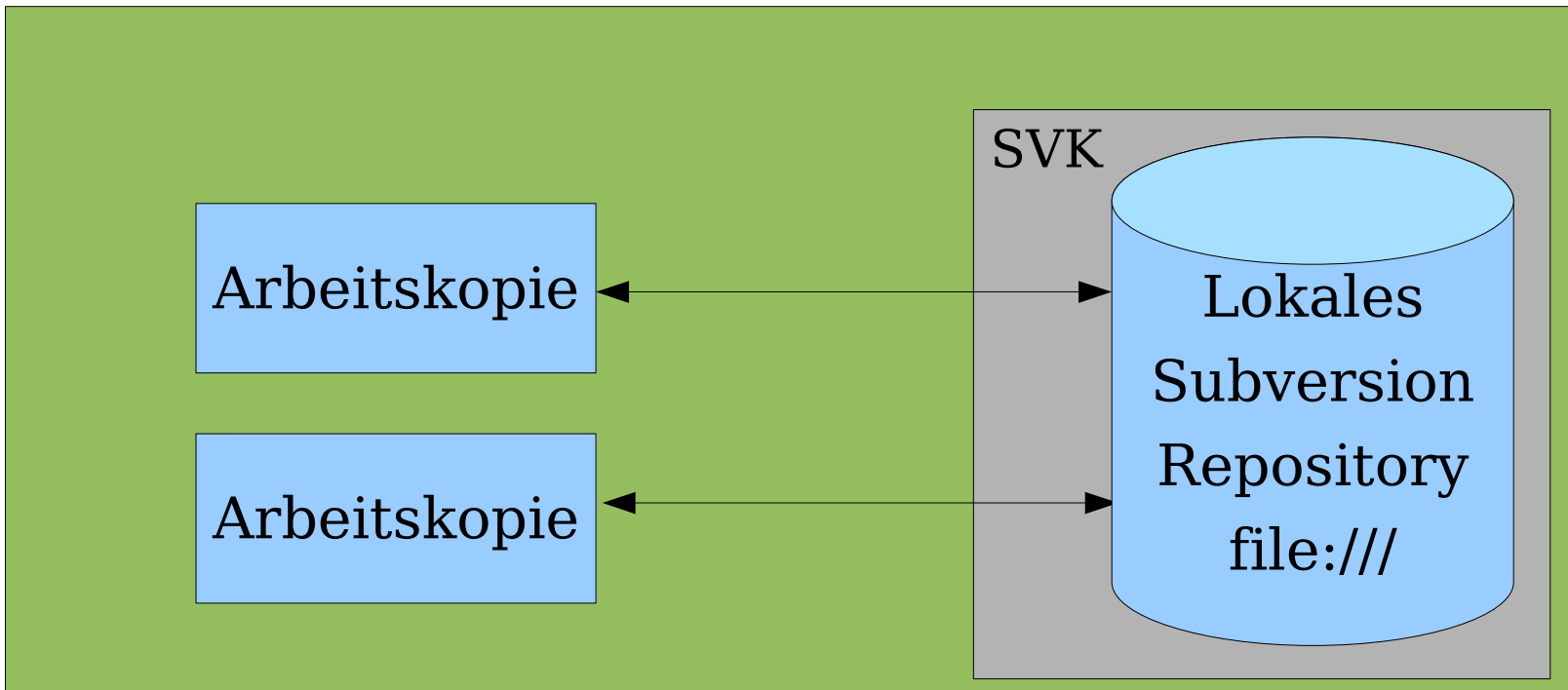
Remote Änderungen

- Die Änderungen können sehr schnell mithilfe
`svn pull` .
- in die aktuell ausgecheckte Arbeitskopie übernommen werden.

Best Practice

- Dem Mitarbeiter wird im Remote Projekt ein eigener Branch zur Verfügung gestellt.
- Der Nachteil ist, dass der Branch mit dem “trunk” gemerged werden muss.

Konzept von SVK



Vorteile von SVK

- Es kann lokal ein- und ausgecheckt werden. Die Historie wird aufgezeichnet.
- Es kann unabhängig weiterentwickelt werden.

Vorteile von SVK

- Keine Speziellen Verzeichnisse bzw. Dateien in den Arbeitskopien wie z.B. bei Subversion oder CVS.
- Soll teilweise schneller sein bei bestimmten Operationen (status etc.)

Nachteile von SVK

- Keinerlei Werkzeugunterstützung in Form von PlugIn's vorhanden.
 - z.B. für Eclipse, IntelliJ, Zend Studio etc.
 - Ausnahme: Ansätze für Unterstützung in Ant.

Nachteile von SVK

- Keinerlei grafische Clients vorhanden, sondern nur als Kommandozeilen Tool verfügbar.
- Eine (un)vollständige Dokumentation.

Online Quellen

- [1] Homepage SVK mit Wiki
 - <http://svk.elixus.org>
- [2] Buch zu SVK (In Arbeit)
 - <http://svkbook.elixus.org>
- [3] Homepage von Subversion
 - <http://subversion.tigris.org>
- [4] Buch zu Subversion
 - <http://svnbook.red-bean.com>

Online Quellen II

- [5] Subversion Forum
 - <http://www.svnforum.org>
- [6] Win32 Installation (Unofficial)
 - <http://home.comcast.net/~klight/svk/>
- [7] Forum für Software Konfigurations Management
 - <http://www.openbc.com/net/skm>

Online Quellen III

- Andere Clients / Bibliotheken etc. für Subversion
 - <http://subversion.tigris.org/links.html>
- Vergleich von SCM Tools
 - <http://better-scm.berlios.de>

Literatur I

- Programmieren unter Linux 2/2006
„Einführung von Subversion in eine Arbeitsgruppe“
 - <http://www.proglinux.org/de>
- Software Developers Journal 3/2006
„Umstieg von CVS auf Subversion“
 - <http://www.sdjournal.org>

Literatur II

- Programmieren unter Linux 3/2006
„Projektmanagement mit SKM“
 - <http://www.proglinux.org/de>
- Vortrag auf den Chemnitzer Linux
Tagen 2006 „Einführung in Subversion“
 - [Einfuehrung Subversion](#)

Noch Fragen?

- Fragen?
 - froscon@soebes.de

- Vielen Dank für Ihre Aufmerksamkeit.