

Versionskontrolle

Richard Madsack

Einleitung

Versionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Zusammenfassung

Versionskontrolle

Verteilte Versionskontrollsysteme

Richard Madsack

25. November 2008

Outline

Versionskontrolle

Richard Madsack

Einleitung

Verisionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Zusammenfassung

- 1 Einleitung
- 2 Verisionskontrolle, wie man sie kennt
- 3 Verteilte Versionskontrollsysteme
- 4 Über den Tellerrand hinaus
- 5 Zusammenfassung

Versionskontrolle?

Versionskontrolle

Richard Madsack

Einleitung

Ein paar Worte zu
Versionskontrolle

Wieso sollte man ein
VCS benutzen?

Versionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Zusammenfassung

Informationen, die ein VCS zur Verfügung stellt

- Was wurde geändert? (diff)
- Wer hat geändert?
- Wann wurde die Änderung durchgeführt?
- (Wieso?)

Vorteile von Versionskontrolle

Versionskontrolle

Richard Madsack

Einleitung

Ein paar Worte zu
Versionskontrolle

Wieso sollte man ein
VCS benutzen?

Versionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Zusammenfassung

Vorteile

- Backup und Wiederherstellung
- Synchronisation
- Undo (kurz und längerfristig)
- Kollaboratives Arbeiten
- Branching (und Merging)
- "Sandboxing"

Versionskontrolle, wie man sie kennt

Versionskontrolle

Richard Madsack

Einleitung

Verisionskontrolle,
wie man sie kennt

Kollaboratives
Arbeiten an einer
Datei...

... und Probleme, die
dabei auftreten
können

Nachteile der
Client-Server
Architektur

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Zusammenfassung

Grundlegende Funktionsweise

- Es existiert *ein* Respository-Server (eine Art File-Server)
- Der Server “merkt sich” *jede* Änderung an einer Datei oder am Verzeichnisbaum
- Aktuelle Version des Servers ist Referenz
- Ein Client kann die aktuelle Version eines Projekts *auschecken*, Änderungen daran vornehmen und diese publizieren, indem er seine Arbeitskopie an den Server “committed”
- Andere Clients “updaten” ihre Arbeitskopie, so werden die Änderungen von dritten automatisch in die Arbeitskopie eingepflegt

Kollaboratives Arbeiten an einer Datei...

Versionskontrolle

Richard Madsack

Einleitung

Versionkontrolle,
wie man sie kennt

Kollaboratives
Arbeiten an einer
Datei...

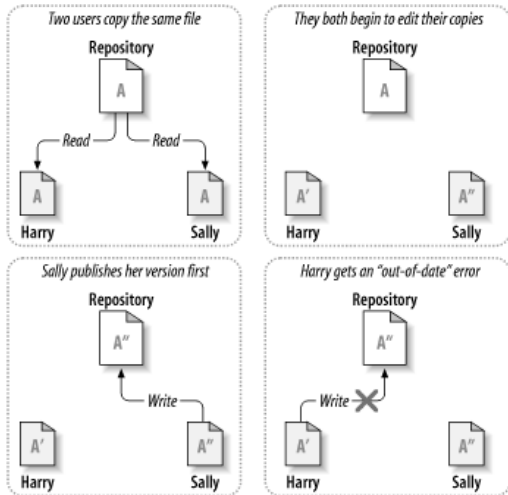
... und Probleme, die
dabei auftreten
können

Nachteile der
Client-Server
Architektur

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Zusammenfassung



... und Probleme, die dabei auftreten können

Versionskontrolle

Richard Madsack

Einleitung

Versionskontrolle, wie man sie kennt

Kollaboratives Arbeiten an einer Datei...

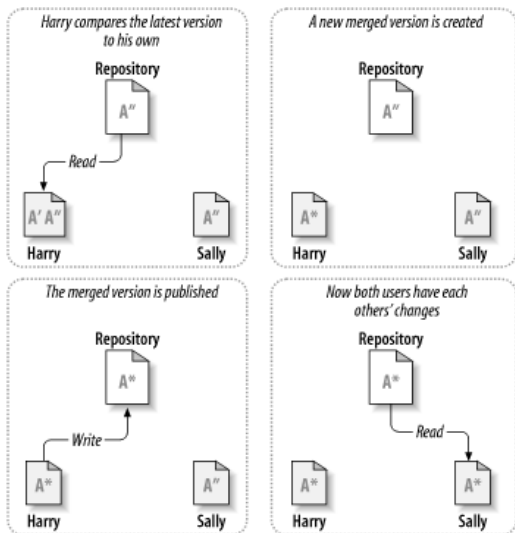
... und Probleme, die dabei auftreten können

Nachteile der Client-Server Architektur

Verteilte Versionskontrollsysteme

Über den Tellerrand hinaus

Zusammenfassung



Nachteile der Client-Server Architektur

Versionskontrolle

Richard Madsack

Einleitung

Versionskontrolle,
wie man sie kennt

Kollaboratives
Arbeiten an einer
Datei...

... und Probleme, die
dabei auftreten
können

Nachteile der
Client-Server
Architektur

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Zusammenfassung

Nachteile

- Abhängigkeit von einer einzelnen physikalischen Maschine
- Client muss sich immer erst mit dem Server synchronisieren, wenn er Änderungen vornehmen möchte
- Eine Netzwerkverbindung zum Server ist absolut notwendig, um am Projekt arbeiten zu können
- Metadaten über den gesamten Dateibaum verteilt (.svn)
- (Hemmung Dinge auszuprobieren)

Verteilte Versionskontrollsysteme

Versionskontrolle

Richard Madsack

Einleitung

Versionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Vorteile von DVCS
Nachteile von DVCS
Noch nie was von
gehört, ist das neu?
Überblick

Über den
Tellerrand hinaus

Zusammenfassung

Distributed VersionControlSystems

- Jeder Entwickler arbeitet auf seinem eigenen lokalen Repository
- Repositories können *gecloned* werden
- Für die wenigsten Operationen ist eine Netzwerkverbindung notwendig
- Es kann viele “zentrale” Repositories geben
- “Lieutenants” (Hierarchie)

Vorteile von DVCS

Versionskontrolle

Richard Madsack

Einleitung

Verionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Vorteile von DVCS

Nachteile von DVCS

Noch nie was von
gehört, ist das neu?

Überblick

Über den

Tellerrand hinaus

Zusammenfassung

Vorteile

- Der Entwickler ist nicht auf eine Netzwerkverbindung angewiesen, um effektiv arbeiten zu können
- Die meisten Operationen sind deutlich schneller, da sie lokal ausgeführt werden
- Keine Abhängigkeit von einer einzelnen physikalischen Maschine
- Der Code kann vom Entwickler zum Bestimmungsort verschiedene Wege durchlaufen
- (Hemmung etwas auszuprobieren sinkt)

Nachteile von DVCS

Versionskontrolle

Richard Madsack

Einleitung

Verisionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Vorteile von DVCS

Nachteile von DVCS

Noch nie was von
gehört, ist das neu?
Überblick

Über den
Tellerrand hinaus

Zusammenfassung

Nachteile

- Es muss mehr *gemergt* werden
- Jeder Entwickler muss die Infrastruktur des “DVCS-Netzes” kennen

Entwicklung des DVCS-Trends

Versionskontrolle

Richard Madsack

Einleitung

Versionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Vorteile von DVCS

Nachteile von DVCS

Noch nie was von
gehört, ist das neu?

Überblick

Über den
Tellerrand hinaus

Zusammenfassung

“Geschichte”

- Bis 2005 wurde zur Entwicklung des Linux-Kernels BitKeeper als VCS eingesetzt
- BitKeeper stellte keine weitere kostenfreie Version zur Verfügung
- Es wurde nach Alternativen gesucht; die existierenden Versionskontrollsysteme wurden den Anforderungen nicht gerecht
- Viele neue Softwareprojekte - Ziel: Verteiltes Versionskontrollsystem für OS-Gemeinde
- Linus Torvalds - GIT
- Matt Mackall - Mercurial

Überblick über aktuelle OpenSource DVCS

Versionskontrolle

Richard Madsack

Einleitung

Verionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Vorteile von DVCS

Nachteile von DVCS

Noch nie was von
gehört, ist das neu?

Überblick

Über den

Tellerrand hinaus

Zusammenfassung

Überblick

- ArX (GNU arch)
- Bazaar
- Codeville
- Darcs
- DCVS
- Fossil
- Git
- Mercurial
- Monotone
- SVK

Monotone

Versionskontrolle

Richard Madsack

Einleitung

Versionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Monotone

Bazaar

Darcs

Git

SVK

Mercurial

Zusammenfassung

die graue Maus

- aktuell Version 0.41
- technische Grundlage: embedded SQLite
- implementiert in C++
- je eine Repositorydatei (=Datenbank) pro Projekt
- Branches und Revisionen müssen global einzigartige Namen haben (SHA1-Hash)
- eigenes *netsync*-Protokoll
- eigene Benutzerverwaltung (RSA-Signaturen)

Bazaar

Versionskontrolle

Richard Madsack

Einleitung

Verionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Monotone

Bazaar

Darcs

Git

SVK

Mercurial

Zusammenfassung

Das Ubuntu-VCS

- aktuell Version 1.9
- implementiert in Python
- bietet “zentralen” Operationsmodus
- unterstützt Import von diversen anderen VCS-Formaten
- Befehlssatz sehr CVS-nah

Darcs 2.0

Versionskontrolle

Richard Madsack

Einleitung

Verionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Monotone

Bazaar

Darcs

Git

SVK

Mercurial

Zusammenfassung

Theory of Patches

- Projekt des Physikers David Roundy
- in Haskell implementiert
- Verwaltet einen Baum mit Patches und Abhängigkeiten
- Patches werden bevorzugt per Email ausgetauscht
- interessanter MergeAlgorithmus

Versionskontrolle

Richard Madsack

Einleitung

Versionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Monotone

Bazaar

Darcs

Git

SVK

Mercurial

Zusammenfassung

Linus DVCS

- aktuell Version 1.6.0.3
- implementiert in C, Perl
- 2 Datenstrukturen: Index, Objekt-Datenbank
- Objekte: blob, tree, commit, tag
- “bläht” sich schnell auf, User muss aufs *packen* achten
- Usability oft kritisiert, starte Verbesserung in den letzten Monaten

Versionskontrolle

Richard Madsack

Einleitung

Versionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Monotone

Bazaar

Darcs

Git

SVK

Mercurial

Zusammenfassung

distributed SVN

- aktuell Version 2.0.2
- implementiert in Perl
- inoffizieller “Nachfolger” von SVN
- arbeitet mit SVN-Dateisystem
- SVN-Repository \Leftrightarrow SVK-Repository

Mercurial

Versionskontrolle

Richard Madsack

Einleitung

Verionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Monotone

Bazaar

Darcs

Git

SVK

Mercurial

Zusammenfassung

hg

- aktuell Version 1.0.2
- implementiert in Python (diff in C)
- intuitiv zu benutzen, schnell
- gute Unterstützung für Dritt-Programme (Merge, Diff)
- wird von vielen großen OpenSource Projekten verwendet: NetBeans, Mozilla, OpenSolaris, uvm.
- IDE-Support für NetBeans/Eclipse

Zusammenfassung

Versionskontrolle

Richard Madsack

Einleitung

Verionskontrolle,
wie man sie kennt

Verteilte Versions-
kontrollsysteme

Über den
Tellerrand hinaus

Zusammenfassung

Fazit

- Es gibt WIRKLICH Alternativen zu SVN/CVS
- Auch für den Eigengebrauch lohnt es sich ein DVCS zu benutzen
- Viele der DVCS sind sehr einfach und intuitiv zu benutzen
- Die Entwicklung geht sehr schnell voran