

git - eine kleine Einfuehrung

Nico Schottelius

SyGroup

2006-02-09

Ziel des Vortrags

- ◆ Was ist git?
- ◆ Wie nutzt man git?
- ◆ Kurzer Geschichtlicher Abriss
 - ◆ Was fuer Alternativen gi(b)t es?

Was ist git?

- ◆ git ist
 - ◆ dumm.
 - ◆ ein Programm.
 - ◆ eine Methode, Daten auf der Festplatte zu speichern.
 - ◆ die Kurzform fuer “gnu interactive tools”
 - ◆ ein (dezentrales) **Versionskontrollsystem**.

Versionskontrolle

- ◆ Speichern von Änderungen
- ◆ Auffinden von Änderungen
- ◆ Änderungen rückgängig machen
- ◆ Arbeit von mehreren Entwicklern koordinieren
- ◆ **Historie** ermöglichen
 - ◆ im Gegensatz zur *Hysterie*

zentral versus dezentral

- ◆ (k)ein zentraler Server
- ◆ Entwicklung abgekoppelt moeglich?
- ◆ Parallelentwicklung
- ◆ zusammenfuegen mehrerer Zweige
- ◆ dezentral ermoeoglicht unabhaengigere Entwicklung
- ◆ Anarchie versus Diktatur

Wer ist noch da?

- ◆ RCS – Revision control system
- ◆ CVS – Concurrent Version system
- ◆ monotone – sehr simpel, c++ und sqlite, leicht traeege
- ◆ tla (aka arch) – kompliziert, schnell, fsdb
- ◆ viele andere

Vorderendprogrammuebersicht

- ◆ cogito (Kommandozeile)
- ◆ gitweb (Webzeile)
- ◆ gitk (viele bunte Zeilen)
- ◆ StGit (ungetestet)

An die Arbeit

- ◆ Wir haben etwas, das wir verwalten wollen
- ◆ Arbeiten mit cogito (**cg**)
- ◆ Initialisieren des Verzeichnisses
 - ◆ **cg-init**
- ◆ Status oder Log anzeigen
 - ◆ **cg-status**
 - ◆ **cg-log**

Bei der Arbeit

- ◆ Etwas aendern
 - ◆ Status anzeigen: **cg-status**
 - ◆ Unterschied anzeigen: **cg-diff**
 - ◆ Aenderung gitifizieren: **cg-commit**
- ◆ Datei kaputt (oder "Alles kaputt") / geloescht
 - ◆ Wiederholen: **cg-revert**

Teilen und verbreiten

- ◆ Zweig (Branch) hinzufuegen
 - ◆ **cg-branch-add** *<name>* *<ziel>*
 - ◆ Lokale oder Ferne ziele moeglich
- ◆ Hochladen, herunterladen, zusammenfuegen
 - ◆ **cg-push** *<name>*
 - ◆ **cg-pull** *<name>*
 - ◆ **cg-merge** *<name>* oder **cg-update** (pull und merge)

etwas huebscher

- ◆ gitweb
 - ◆ Webzugriff, Perl
 - ◆ verschiedene Projekte, einfache Uebersicht
- ◆ gitk
 - ◆ graphisch (X11), Python
 - ◆ leichte Analyse der Aenderungen eines Projektes
- ◆ cg-log -s -c
 - ◆ Farbliche (bunte) Logausgabe / Zusammenfassung

Individualisierung

- ◆ Anpassen des Editors
 - ◆ EDITOR=ooffice cg-commit
- ◆ Namen / E-Post-Adresse aendern
 - ◆ GIT_AUTHOR_NAME
 - ◆ GIT_AUTHOR_EMAIL
- ◆ weitere Optionen moeglich

Quellen

- ◆ <http://git.or.gz/> - Heimatseite von git
- ◆ <http://www.kernel.org/pub/software/scm/git/docs/tutorial.html> - Eine Anleitung

Zusammenfassung

- ◆ git wird meist durch cogito benutzt
- ◆ git ist schnell, relativ einfach und komfortabel
- ◆ grosse Entwicklergemeinde
 - ◆ schnelle Behebung von elektronischen Käfern
- ◆ dezentral organisiert
- ◆ verschiedene Programme zur Visualisierung und Bearbeitung

Fragen?