

# NoSQL Datenbanken

Labor 09.10 - Hochschule Mannheim

# Repository

- Verwaltetes Verzeichnis zur Speicherung von Daten
- Versionsverwaltungssysteme von denen ausgehend:
  - *eingchecked* und
  - *ausgecheckt* wird.
- Veränderungen werden protokolliert.
- Rekonstruktion von früheren Zuständen.
- Konfliktbearbeitung?!
- Meist auch mit Metadaten für einen Paketmanager.

# Subversion (SVN)

- Zentralisiertes Versionsverwaltungssystem
- 4 Jahre Entwicklung - Version 1.0 am 23.2.2004
- Entwickelt von CollabNet
- Seit dem 10.2.2010 ein Apache Top-Level Projekt
- Freies / Open Source Versionskontrollsystem
- verwaltet Dateien und Verzeichnisse und deren Änderungen.

Literatur:

<http://svnbook.red-bean.com/>

# Begriffe

Revision

Changeset

Delta / Diff

Merge

Branch

Tag

# Revision

- Synonym für den Begriff Version
- Entwicklungsstadium der kompletten Software
- Versionsnummern

# Changeset

- Zusammenhang von Änderungen einer Version.
- Notation, die die Operation mit / auf dem Bestandteil des Changeset erklärt:
  - U = Updated
  - D = Deleted
  - A = Added
- Bsp.:
  - U foo.txt
  - A katze.php
  - D wichtigeDateiDoNotDelete.txt

# Delta / Diff

- Griechisch Delta ( $\Delta$ ) für Benennung von Differenzen
- In SVN: angewandte Speicherverfahren
- Es werden immer nur die Unterschiede zwischen zwei Versionen festgehalten.

# Tag

- Beschriftung einer bestimmten, einzelnen Revision.
- Markierung eines definierten Standes mit einem Zeiger.
- Häufige Verwendung für die Definition von Versionen.



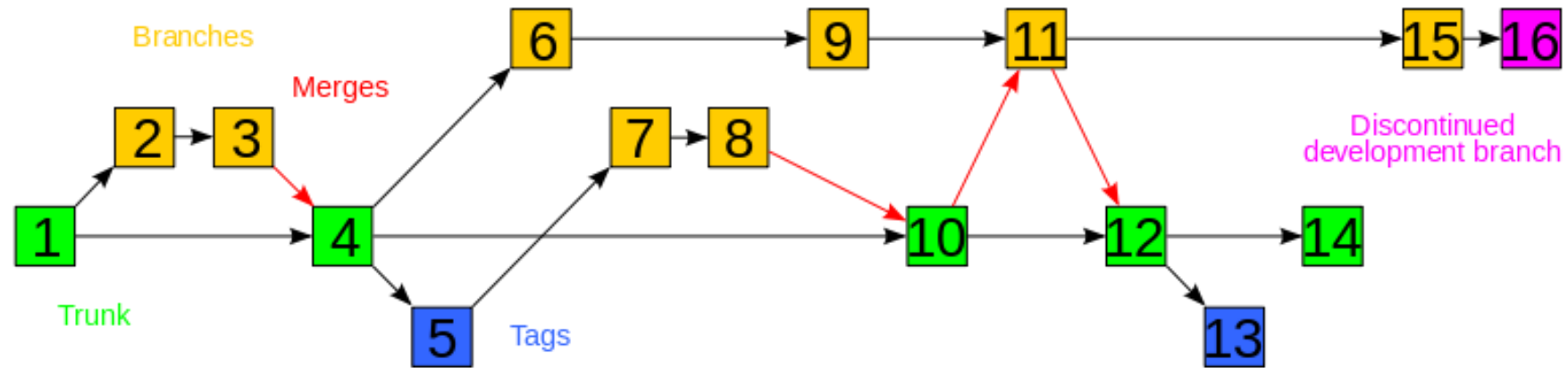
# Branch

- (= Ast) eine Ab-/Verzweigung einer Hauptentwicklungslinie (meist „trunk“ genannt).
- In SVN: Kopien von einer Ursprungsversion
- Trennung verschiedener Versionsstände
- Trennung verschiedener Features (Featurebranch)

# Merge

- Zusammenführen von verschiedenen Änderungen in zwei Versionen einer Datei.
- Mergen von zwei Branches.

# SVN Timeline



Quelle: [http://en.wikipedia.org/wiki/File:Subversion\\_project\\_visualization.svg](http://en.wikipedia.org/wiki/File:Subversion_project_visualization.svg)

# Vorteile

- Kostenfrei
- Erprobt im Alltag
- Stetige Entwicklung als Apache Project
- Unterstützung von vielen IDEs
- Einfach in der Handhabung
- Weit verbreitet

# Einschränkungen

- Es werden nur Deltas verwaltet
- Ohne Server geht nichts
- Automatisches Mergen ist keine schöne Erfahrung. (= viele Konflikte bei offensichtlich eindeutigen Situationen).

# Git

- dezentralisiertes Versionsverwaltungssystem
- Erfunden von Linus Torvalds
- Gestartet April 2005
- Aktuelle Version 2.6 (stable Release)
- Besonderheit: Merge Gedächtnis:
  - Git kennt seine Elternknoten.

Literatur:

<http://git-scm.com/book/de>

# Begriffe

Pull

Staging

Commit

Dirty

Push

Clone

# Clone

- Spiegeln einer vollständigen Historie in ein (lokales) Repository.
- Inklusive Commits, Tags und Branches.



# Staging

- Hinzufügen von Dateien in einen virtuellen Bereich
- Daten im Stage kommen in den nächsten Commit

# Commit

- Übermittelt den veränderten Inhalt zu dem lokalem Repo
- Änderung sind noch nicht im Remote Repo

# Push

- Übermittelt den Inhalt eines Branches aus einem lokalem Repo an ein Remote Repo
- Transferiert Commit-By-Commit

# Pull

- „Zieht“ Änderungen aus einem Remote Repo
- Sämtliche Branches werden ebenfalls mit “gezogen“.

# Dirty

- Branches die,
  - Änderungen ggü. dem letzten Commit besitzen
  - Änderungen noch nicht (vollständig) im Staging haben
- Bsp.:
  - Foo.txt (tracked, unchanged)
  - Katze.php (tracked, changed, unstaged)

# Vorteile

- Schnell, da Vielzahl der Operationen lokal
- Unabhängig, da kein Server nötig.
- Sicher, da jeder alles besitzt.
- Weniger Frust bei dem Mergen

# Einschränkung

- Komplizierter Einsatz unter Windows
- Kaum GUIs oder IDE Plugins

# Zusammenfassung

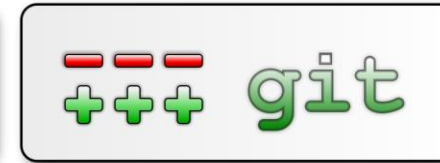


- Etabliert
- Techn. Ausgereift
- Verstanden
- Unterstützt
- Verfügbar

- Langsam
- Historie „dumm“
- Anstrengendes Branching
- Ohne Server unbrauchbar



- Schnell
- Mergen ist sicher & einfach
- Branching ist erwünscht
- Ohne Server verwendbar
- Historie „schlau“



- Steiniger Einstieg
- Verlangt einen Paradigmenwechsel
- Nicht aus allen OS „gut“



# Repository anlegen

## ▶ **Git:**

<https://github.com/>

<https://bitbucket.org> (privat)

`git clone [url]`

## ▶ **SVN:**

<http://projectlocker.com/>

## ▶ graphische Benutzeroberfläche:

<http://tortoisesvn.net/>

# Regeln für das Repository

- Benennungsschema (Repo):  
    <GruppenName>  
    Bsp: [https://github.com/CoolGruppeONE/\(..\).git](https://github.com/CoolGruppeONE/(..).git)

# Commit-Messages

- **Changed:** foo.html, h2-Tag Task 1.1 - mkraus
  - ▶ Dateiname,
  - ▶ Was wurde geändert?,
  - ▶ Zu welcher Aufgabe gehörte die Änderung?,
  - ▶ Wer war daran beteiligt?
- **Added:** Neue Datei hinzugefügt.
- **Changed:** Vorhandene Datei verändert.
- **Deleted:** Datei gelöscht.