



git



HOCHSCHULE
FÜR ANGEWANDTE
WISSENSCHAFTEN · FH
MÜNCHEN

Versionsverwaltung mit Git

Eine Einführung ins Source Code Management

Agenda

- **Einführung**
 - Beispiele für Version Control Systems
 - Geschichte und Entwicklung von Git
 - Designziele bei Git
- **Wie Git funktioniert ...**
- **Wichtige Tools**
- **Arbeiten mit „remotes“**
- **Ausblick**



Beispiele für Version Control Systems

- CVS
- Subversion/SVN
- Mercurial (hg)
- Git



Geschichte und Entwicklung von Git

- Linux-Kernel brauchte neues VCS
- April 2005: Linus Torvalds veröffentlicht erste Version von Git
- Dezember 2005: Version 1.0.0

- Aktuell: Version 1.8.0

„I'm an egoistical bastard, and I name all my projects after myself.
First Linux, now git.“ – Linus Torvalds



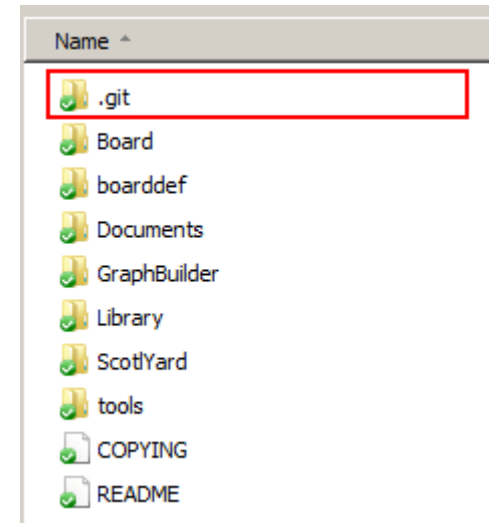
Designziele bei Git

- Starke Unterstützung für
 - Nicht-lineare Entwicklung
 - Verteilte Arbeitsabläufe
- Sicherheit gegen Verfälschung
- Effizienz



Wie Git funktioniert ...

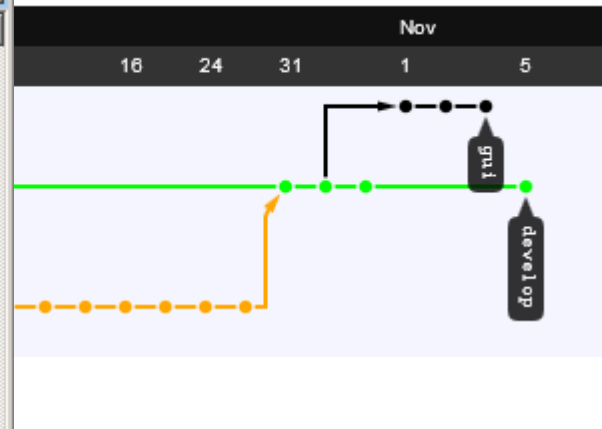
- Repository
- Branch
- Commit (Checkin)



```
MINGW32:/d/Daten/Dropbox/Projekte/ScotlYard/gitrep
jakob190590@HELIO /d/Daten/Dropbox/Projekte/ScotlYard/gitrep (develop)
$ git show f95f3
commit f95f3ae83958a721f38bd74cc4c04d0740c28a82
Author: Jakob Schöttl <jschoett@gmail.com>
Date:   Mon Sep 17 22:45:22 2012 +0200

    Bugfix in Core: Add instead of remove!!

diff --git a/ScotlYard/src/kj/scotlyard/game/model/DefaultGame.java b/ScotlYard/
index e311f1a..00ddc46 100644
--- a/ScotlYard/src/kj/scotlyard/game/model/DefaultGame.java
+++ b/ScotlYard/src/kj/scotlyard/game/model/DefaultGame.java
@@ -109,7 +109,7 @@ public class DefaultGame extends AbstractGameState implement
    @Override
    public DetectivePlayer remove(int index) {
        DetectivePlayer p = list.remove(index);
-       getPlayerListenerInformer().detectiveAdded(DefaultGame.t
+       getPlayerListenerInformer().detectiveRemoved(DefaultGame
        return p;
    }
}
```



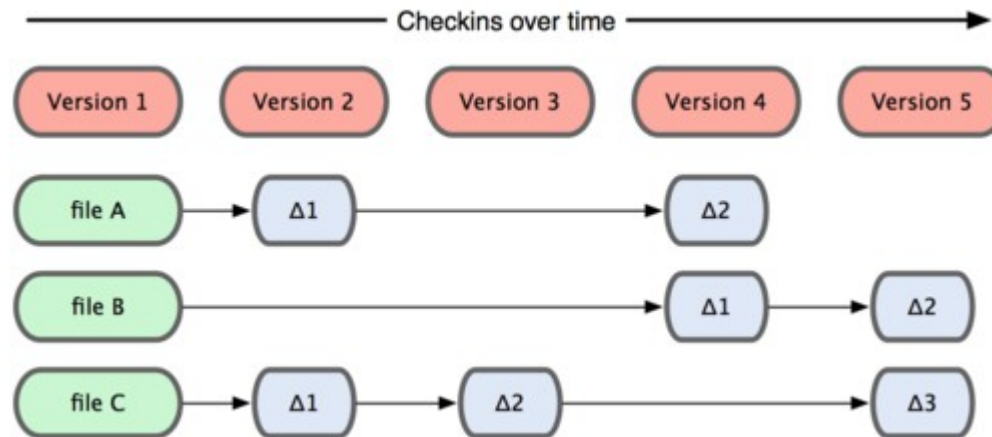
18.11.12

Wie Git funktioniert ...

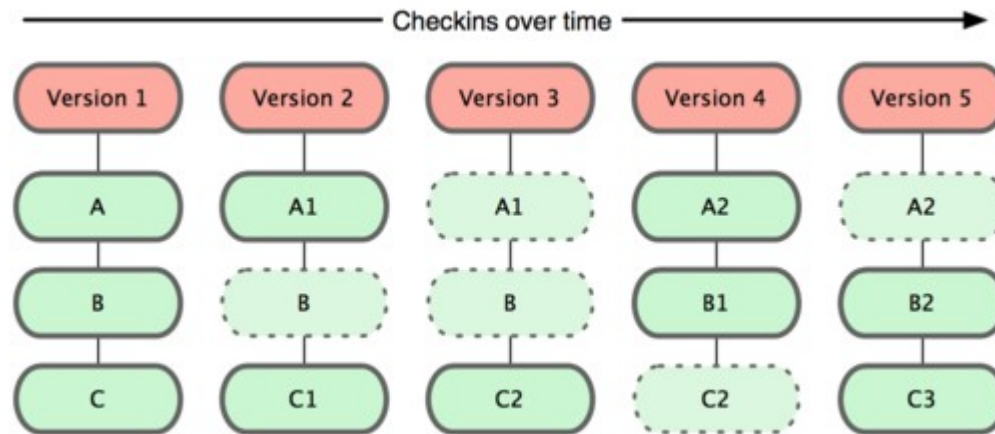
- **Repository**
- Branch
- Commit



Wie Git funktioniert ...



Wie Git funktioniert ...



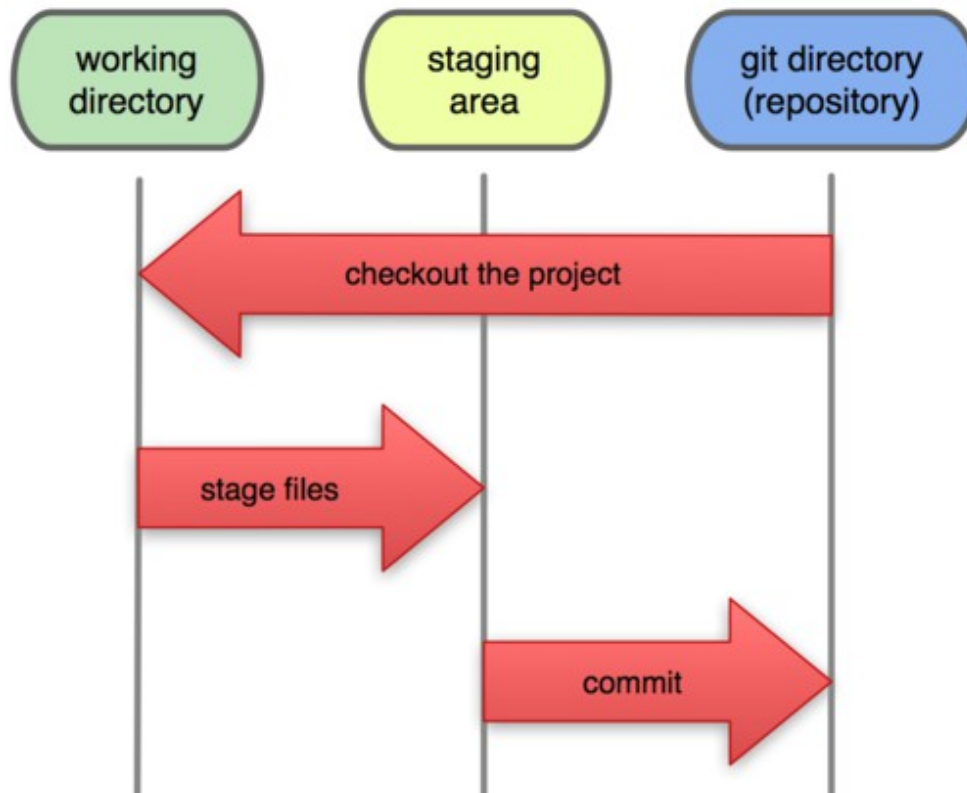
Wie Git funktioniert ...

- **Repository**
- Branch
- **Commit**

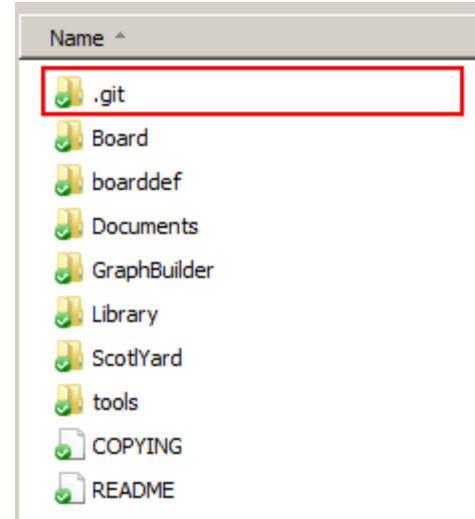


Wie Git funktioniert ...

Local Operations



- Add
- Remove (rm)
- Commit

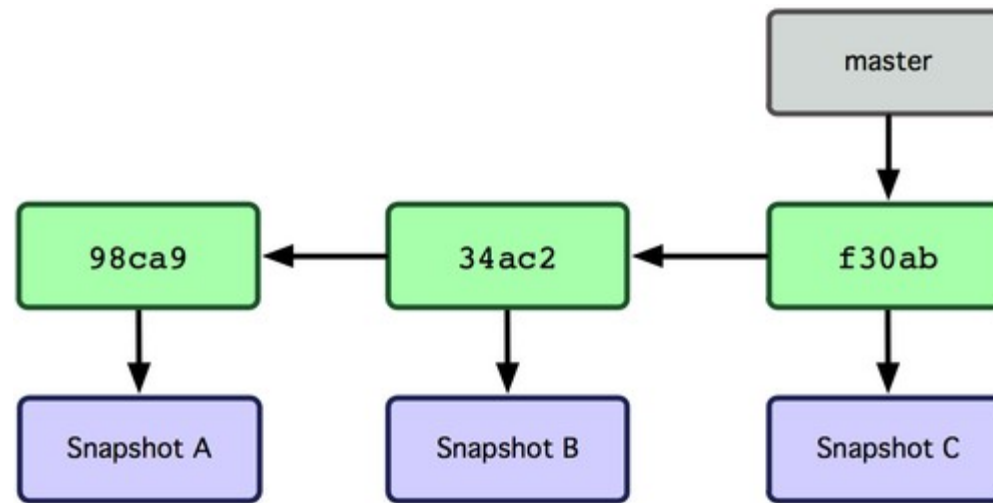


Wie Git funktioniert ...

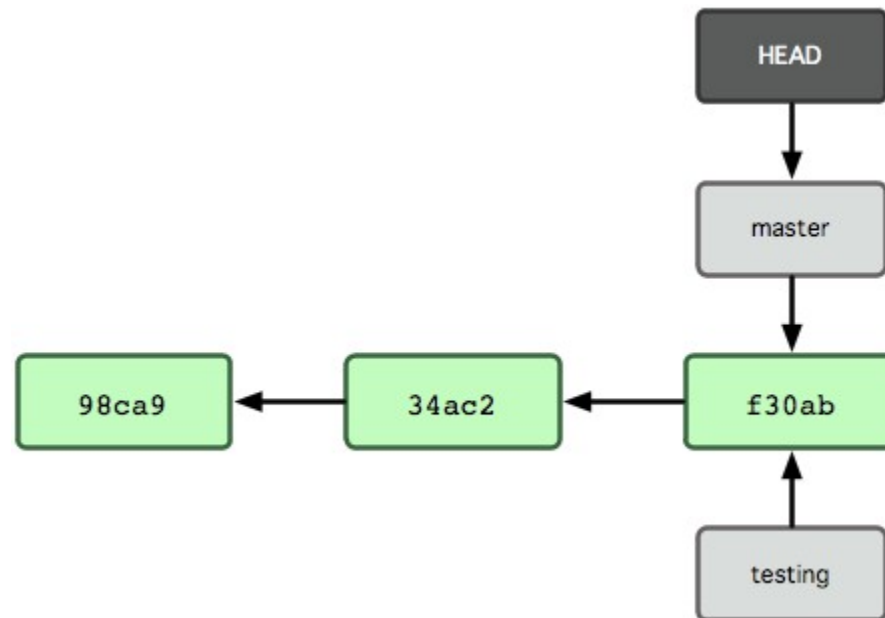
- Repository
- **Branch**
- Commit (Checkin)



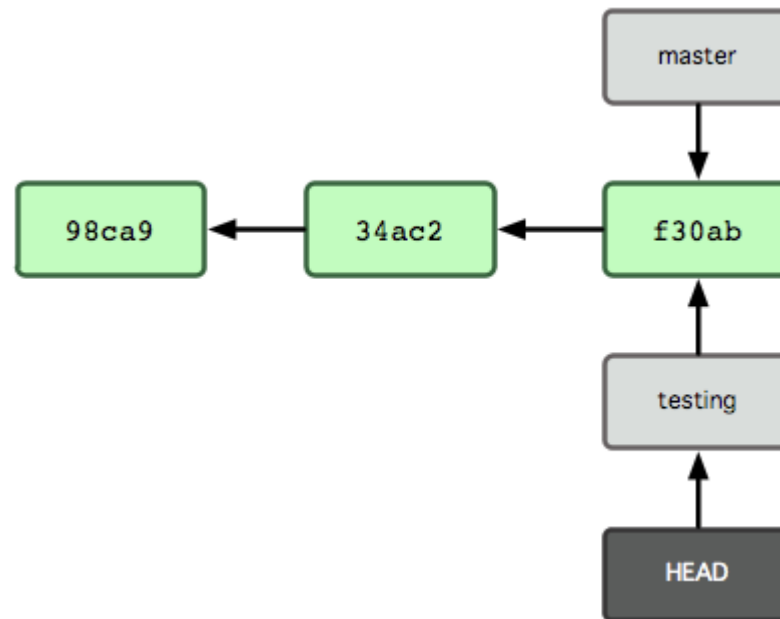
Wie Git funktioniert ...



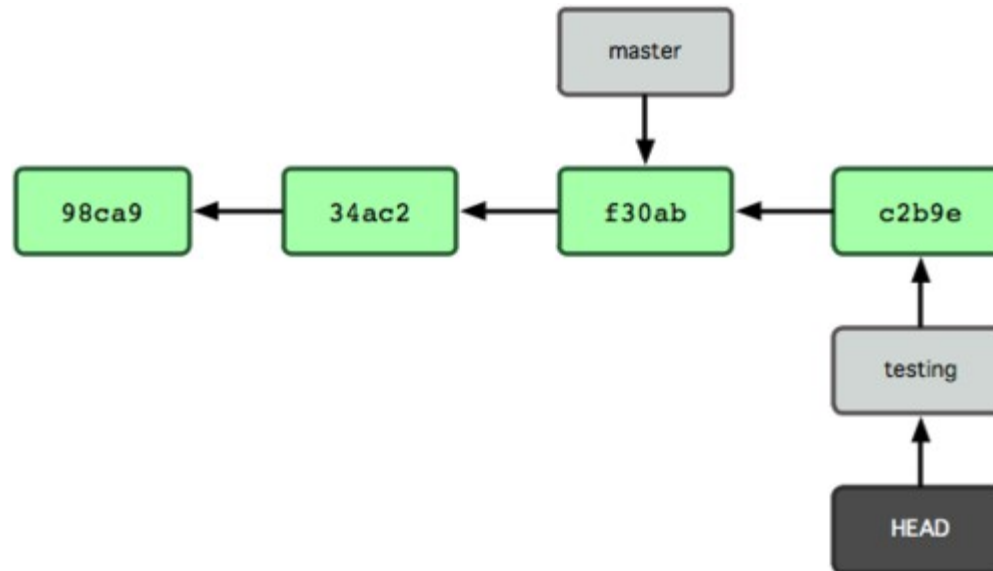
Wie Git funktioniert ...



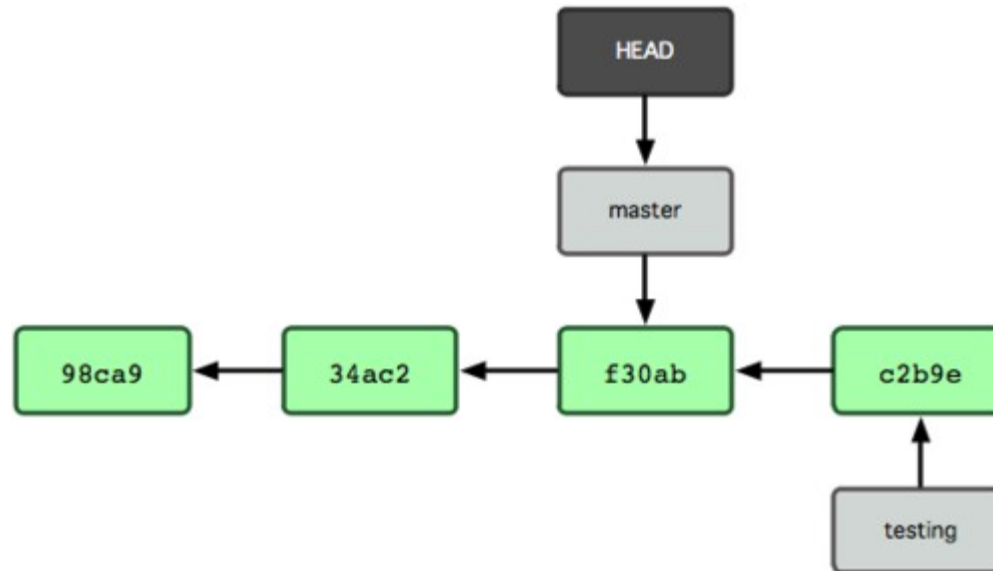
Wie Git funktioniert ...



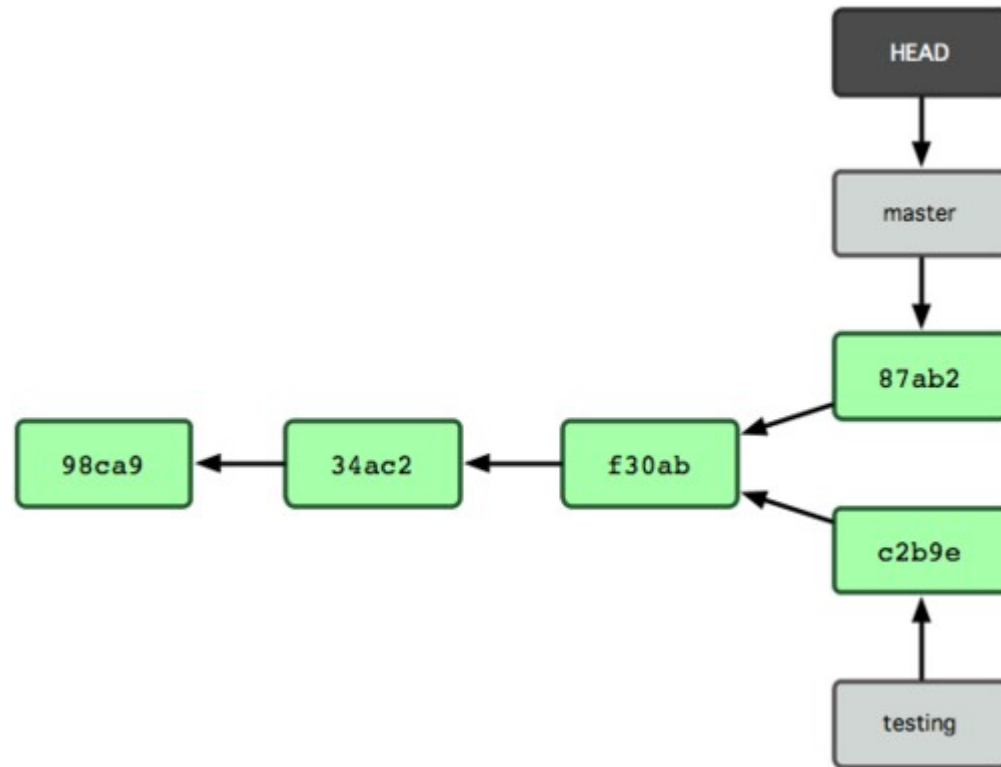
Wie Git funktioniert ...



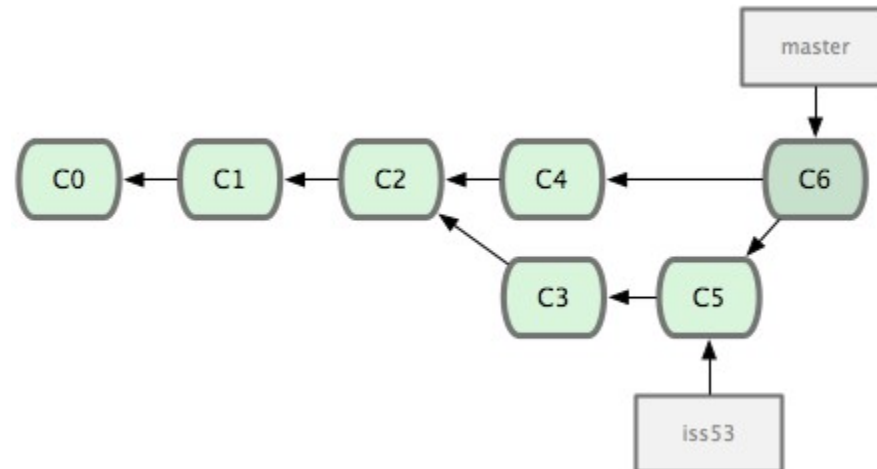
Wie Git funktioniert ...



Wie Git funktioniert ...



Wie Git funktioniert ...



Wichtige Tools

- Log
- Show
- Diff



Arbeiten mit „remotes“

- „remotes“ sind andere Git-Repositories
- Clone
- Push
- Fetch (Pull)



Ausblick

- Undoing
- Stashing
- Tagging
- Rebasing
- Workflows
- Submodules
- Customization
- ...



Danke für die Aufmerksamkeit

Quellen

- <http://git-scm.com>
- Scott Chacon: Pro Git
Aus dem Buch sind speziell auch die tollen Grafiken
- Wikipedia und deren Quellen
- Die Webseiten der anderen VCS
- Mailinglisten

Lizenz: CC BY-NC-SA 3.0 <<http://creativecommons.org/licenses/by-nc-sa/3.0/>>



Anhang (1) – Tipps

- Introduction to Git (1:22 h Video auf YouTube)
<https://www.youtube.com/watch?v=ZDR433b0HJY>
- Das Buch Pro Git gibt's kostenlos auf
<http://git-scm.com/book>
- Hilfe zu einzelnen Git-Befehlen:
`git [command] --help`



Anhang (2) – Git vs. Other VCSs

Git

- Snapshots
 - Sicherer gegen Datenverlust
 - Schneller bei Diffs/Checkouts/...
- Verteilt
 - Unabhängig von Server u. Verbindung
 - Sicherer gegen Datenverlust

Other VCSs

- Deltas
- Zentral

