

Version Control

eine Einführung

Zoran Zarić

- ▶ 2007 Ausbildung zum Fachinformatiker für Systemintegration
- ▶ Informatik Student an der TU Darmstadt seit 2008
- ▶ Git seit Mitte 2009
- ▶ bup seit April 2010

1. Fragen
2. Basics
3. Git
4. Beispiele / Demo
5. Hintergründe
6. Tools

Fragen

Fragen

1. Wer weiß was Version Control ist?

Fragen

1. Wer weiß was Version Control ist?
2. Wer kennt Subversion / CVS?

Fragen

1. Wer weiß was Version Control ist?
2. Wer kennt Subversion / CVS?
3. Wer weiß was *Distributed* Version Control ist?

Fragen

1. Wer weiß was Version Control ist?
2. Wer kennt Subversion / CVS?
3. Wer weiß was *Distributed* Version Control ist?
4. Wer kennt Git / Mercurial?

Version Control?

Fragen

```
$ echo "Halo Welt!" > README
```

```
$ mkdir v0.1
```

```
$ mv README v0.1
```

```
$ echo "Hallo Welt!" > README
```

Distributed Version Control?

Fragen

```
$ echo "Hier die bestimmt finale Version unseres Codes" \  
| mail -s "Code final, echt jetzt" -a main.c foo@bar.com
```

Git

- ▶ “stupid content tracker”
- ▶ “content addressable”
- ▶ Verteiltes Versionierungssystem
 - ▶ Vollständige Kopie des Repositories lokal
 - ▶ lokales Committen, Branchen, Mergen
- ▶ Linus Torvalds begann Entwicklung 2005
- ▶ Snapshot- statt Diff-basiert

Diffs? Snapshots?

Fragen

Diffs? Snapshots?

Diffs (Subversion)

```
$ cat foo.1  
Halo Wlet!
```

```
$cat foo.2.diff  
1c1  
< Halo Wlet!  
---  
> Hallo Welt!
```

```
$patch foo.1 -o - < foo.2.diff  
patching file foo.1  
Hallo Welt!
```

Snapshots (Git)

```
$ cat foo.1  
Halo Wlet!
```

```
$cat foo.2  
Hallo Welt!
```

Installation

- ▶ Windows

mysysgit <http://code.google.com/p/msysgit/>

TortoiseGit <http://code.google.com/p/tortoisegit/>

- ▶ Ubuntu

```
aptitude install git
```

- ▶ OSX

<http://code.google.com/p/git-osx-installer/>

Beispiele

Config

```
$ git config user.name "Zoran Zaric"
```

```
$ git config user.email "zz@zoranzaric.de"
```

Beispiele

Erste Schritte

```
$ git init
```

```
$ echo "Halo Wlet!" > README
```

```
$ git add README
```

```
$ git commit -m "Initial import" # (1)
```

```
$ git log
```

```
$ echo "Hallo Welt!" > README
```

```
$ git add README
```

```
$ git commit -m "Fix spelling errors" # (2)
```

```
$ git log
```



Beispiele

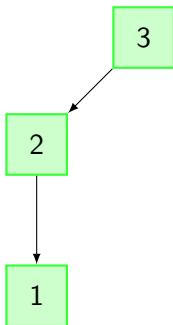
Branchen und Mergen

```
$ git status
# On branch master

$ git checkout -b new-branch
$ git branch

$ echo "Lorem ipsum" >> README
$ git add README
$ commit -m "Add noise" # (3)

$ git diff master
$ git checkout master
$ git merge new-branch
$ git branch -d new-branch
```



Äääähm!

war da nicht was mit *distributed?!1elf*

Beispiele

Remotes

```
$ git clone https://github.com/apenwarr/bup.git
```

```
$ git remote rename origin upstream
```

```
# Änderungen holen
```

```
$ git fetch upstream
```

```
$ git merge upstream/master
```

```
# Änderungen hochladen
```

```
$ git push upstream master
```

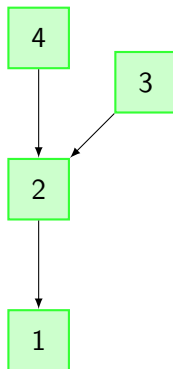
Aber, aber...

upstream sind Änderungen und ich
habe Änderungen... und... und...
jetzt?

Beispiele

Rebase

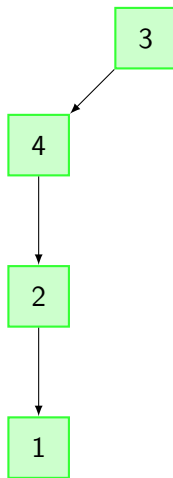
```
# Lokale Änderungen auf Basis  
# von upstream/master  
$ echo "... " >> README  
$ git add README  
$ git commit -m "... " # (3)  
  
# Änderungen holen  
$ git fetch upstream # (4)
```



Beispiele

Rebase

```
# Lokale Änderungen auf Basis  
# von upstream/master  
$ echo "... " >> README  
$ git add README  
$ git commit -m "... " # (3)  
  
# Änderungen holen  
$ git fetch upstream # (4)  
  
# Eigene Änderungen auf Upstream  
# anwenden  
$ git rebase upstream/master
```



Basics

- ▶ SHA1
- ▶ DAG (Directed Acyclic Graph)
gerichteter zyklener freier Graph

Ein Repository

- ▶ BLOBs

e69de29

Hello World

Ein Repository

- ▶ BLOBs

e69de29

- ▶ Trees

82e3a75

100644 blob

5e1c309dae7f45e0f39b1bf3ac3cd9db12e

README

Ein Repository

- ▶ BLOBs

e69de29

- ▶ Trees

82e3a75

- ▶ Commits

3dfe461f

tree

a3d703e579dc9baae20456eb63fa49f5e4e

author Zoran Zaric

<zz@zoranzaric.de>1314498536

+0200

committer Zoran Zaric

<zz@zoranzaric.de>1314498536

+0200

Example commit

Ein Repository

- ▶ BLOBs

e69de29

- ▶ Trees

82e3a75

- ▶ Commits

3dfe461f

- ▶ Tags & Branches

v0.1

master

Ein Repository

- ▶ BLOBs

e69de29

- ▶ Trees

82e3a75

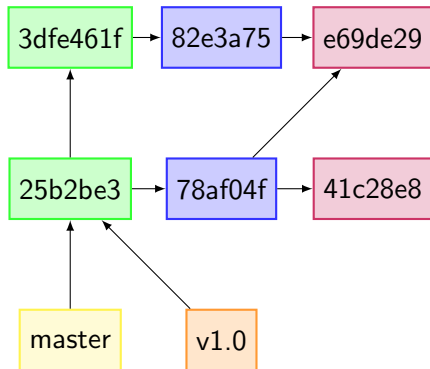
- ▶ Commits

3dfe461f

- ▶ Tags & Branches

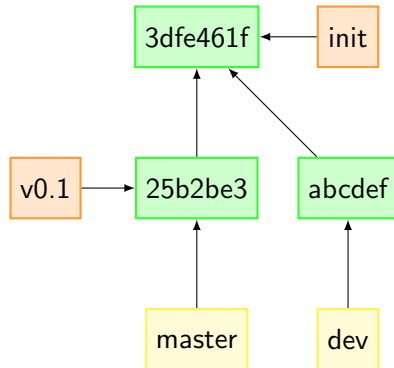
v0.1

master



Ein Repository

- ▶ DAG
Directed Acyclic Graph



Ein Repository

- ▶ DAG
Directed Acyclic Graph
- ▶ Packfiles

e69de29

82e3a75

3dfe461f

41c28e8

78af04f

25b2be3

Tools

Tools

- ▶ Tig
- ▶ gitk
- ▶ gitg
- ▶ Tower Git
- ▶ gitkx

Weiterführendes / Nachschlagewerk

- ▶ <http://progit.org/>
- ▶ <http://gitref.org/>
- ▶ <http://sea.ucar.edu/event/unlocking-secrets-git>
- ▶ <http://blip.tv/scott-chacon>

Danke

- ▶ zorzar auf freenode & hackint
- ▶ zz@zoranzaric.de (Email & Jabber)
- ▶ zoranzaric.de
- ▶ github.com/zoranzaric
- ▶ gplus.zoranzaric.de
- ▶ @zoranzaric

- ▶ Slides: zoranzaric.de/git-opphase1112.pdf