

# Versionkontrolle mit git

Wer bin ich?

Wer bin ich?

Federico Hernandez

Wer bin ich?

Darmstadt

Wer bin ich?

Göteborg, Schweden

Wer bin ich?

Mathematiker

Wer bin ich?

Senior Linux/Unix System Administrator

Wer bin ich?

Linux User seit ...

Wer bin ich?

Ultramarathon Läufer

Wer bin ich?



# Warum Versionskontrolle?

## Warum Versionskontrolle?

```
$ cp file.pl file.pl.old
```

## Warum Versionskontrolle?

```
$ cp file.pl file.pl.old  
$ vi file.pl
```

## Warum Versionskontrolle?

```
$ cp file.pl file.pl.old  
$ vi file.pl  
$ cp file.pl file.pl.0910140935
```

## Warum Versionskontrolle?

```
$ cp file.pl file.pl.old  
$ vi file.pl  
$ cp file.pl file.pl.0910140935
```

```
$ vi file.pl
```

## Warum Versionskontrolle?

```
$ cp file.pl file.pl.old  
$ vi file.pl  
$ cp file.pl file.pl.0910140935
```

```
$ vi file.pl  
$ git commit -a -m "Fixed bug #123"
```

# Versionkontrollsysteme

Lokal: rcs

Lokal: rcs

Zentraler Server: cvs, svn

# Versionkontrollsysteme

Lokal: rcs

Zentraler Server: cvs, svn

Verteilt: git



Bitkeeper

Bitkeeper  
schnell,

Bitkeeper

schnell, einfach,

Bitkeeper

schnell, einfach, nicht-linear,

Bitkeeper

schnell, einfach, nicht-linear, verteilt,

Bitkeeper

schnell, einfach, nicht-linear, verteilt, kryptografisch stark

Bitkeeper

schnell, einfach, nicht-linear, verteilt, kryptografisch stark

Start: 2005-04-03

Bitkeeper

schnell, einfach, nicht-linear, verteilt, kryptografisch stark

Start: 2005-04-03

Ankündigung: 2005-04-06

Bitkeeper

schnell, einfach, nicht-linear, verteilt, kryptografisch stark

Start: 2005-04-03

Ankündigung: 2005-04-06

Selbsttragend: 2005-04-07

Bitkeeper

schnell, einfach, nicht-linear, verteilt, kryptografisch stark

Start: 2005-04-03

Ankündigung: 2005-04-06

Selbsttragend: 2005-04-07

Multibranch merge: 2005-04-18



Snapshots

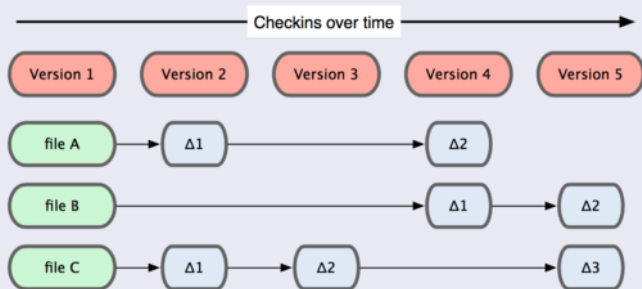


Figure: Scott Chacon - <http://progit.com>

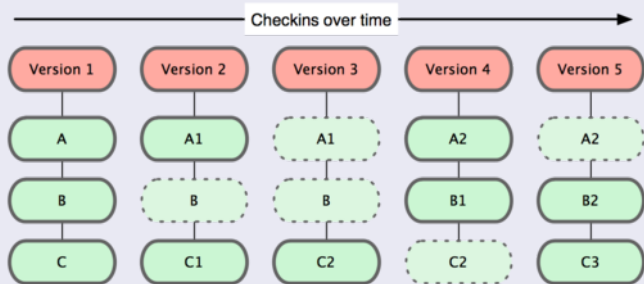


Figure: Scott Chacon - <http://progit.com>

Snapshots

Überwiegend lokale Kommandos

Snapshots

Überwiegend lokale Kommandos

SHA-1

Snapshots

Überwiegend lokale Kommandos

SHA-1

3 lokale Bereiche:

Snapshots

Überwiegend lokale Kommandos

SHA-1

3 lokale Bereiche: Arbeitsverzeichnis

Snapshots

Überwiegend lokale Kommandos

SHA-1

3 lokale Bereiche: Arbeitsverzeichnis, Staging

Snapshots

Überwiegend lokale Kommandos

SHA-1

3 lokale Bereiche: Arbeitsverzeichnis, Staging, Repository

# Zyklus

## Local Operations

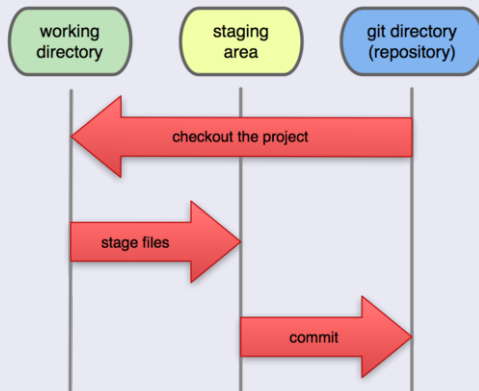


Figure: Scott Chacon - <http://progit.com>

## File Status Lifecycle

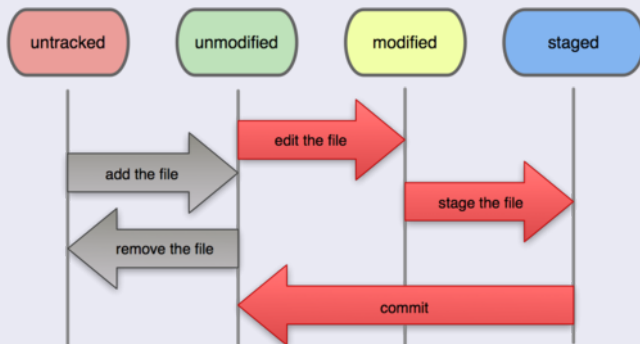


Figure: Scott Chacon - <http://progit.com>

# Installation

# Installation

```
$ sudo apt-get install git-core
```

# Installation

## Installation

```
$ sudo apt-get install curl-devel expat-devel \  
  gettext-devel openssl-devel zlib-devel
```

## Installation

```
$ sudo apt-get install curl-devel expat-devel \  
  gettext-devel openssl-devel zlib-devel
```

Sourcen: <http://www.kernel.org/pub/software/scm/git>

## Installation

```
$ sudo apt-get install curl-devel expat-devel \  
  gettext-devel openssl-devel zlib-devel
```

Sourcen: <http://www.kernel.org/pub/software/scm/git>

```
$ tar zxvf git-1.6.5.tar.gz  
$ cd git-1.6.5
```

## Installation

```
$ sudo apt-get install curl-devel expat-devel \  
  gettext-devel openssl-devel zlib-devel
```

```
Sourcen: http://www.kernel.org/pub/software/scm/git
```

```
$ tar zxvf git-1.6.5.tar.gz  
$ cd git-1.6.5  
$  
$ make configure  
$ ./configure --prefix=/usr/local  
$ sudo make install  
$ cd ..
```

## Installation

```
$ sudo apt-get install curl-devel expat-devel \  
  gettext-devel openssl-devel zlib-devel
```

```
Sourcen: http://www.kernel.org/pub/software/scm/git
```

```
$ tar zxvf git-1.6.5.tar.gz
```

```
$ cd git-1.6.5
```

```
$
```

```
$ make configure
```

```
$ ./configure --prefix=/usr/local
```

```
$ sudo make install
```

```
$ cd ..
```

```
$
```

```
$ sudo tar zxv -C /usr/local/man \  
  -f git-manpages-1.6.5.tar.gz
```

# git Kommandos

150 Kommandos in `/usr/local/libexec/git-core`

150 Kommandos in `/usr/local/libexec/git-core`  
25-30 Day-to-day

150 Kommandos in `/usr/local/libexec/git-core`

25-30 Day-to-day

`git befehl`  $\Rightarrow$  `git-befehl`

# git Kommandos

150 Kommandos in `/usr/local/libexec/git-core`  
25-30 Day-to-day

`git befehl` ⇒ `git-befehl`

`git befehl help` ⇒ `man git-befehl`

# git Kommandos

# git Kommandos

create

init

clone

# git Kommandos

## create

init  
clone

## browse

status log  
diff branch  
tag blame

# git Kommandos

## create

init  
clone

## browse

status log  
diff branch  
tag blame

## track

add commit  
mv rm  
tag

# git Kommandos

## create

init  
clone

## browse

status log  
diff branch  
tag blame

## track

add commit  
mv rm  
tag

## branch

checkout branch  
merge rebase

# git Kommandos

## create

init  
clone

## browse

status log  
diff branch  
tag blame

## track

add commit  
mv rm  
tag

## branch

checkout branch  
merge rebase

## undo

reset checkout  
revert

# git Kommandos

## create

init  
clone

## browse

status log  
diff branch  
tag blame

## track

add commit  
mv rm  
tag

## branch

checkout branch  
merge rebase

## undo

reset checkout  
revert

## update

pull fetch  
fetchmerge  
apply

# git Kommandos

## create

init  
clone

## browse

status log  
diff branch  
tag blame

## track

add commit  
mv rm  
tag

## branch

checkout branch  
merge rebase

## undo

reset checkout  
revert

## update

pull fetch  
fetchmerge  
apply

## publish

push  
format-patch

# git Kommandos

## create

init  
clone

## browse

status log  
diff branch  
tag blame

## track

add commit  
mv rm  
tag

## branch

checkout branch  
merge rebase

## undo

reset checkout  
revert

## update

pull fetch  
fetchmerge  
apply

## publish

push  
format-patch

## config

config  
remote

# git Kommandos

## create

init  
clone

## browse

status log  
diff branch  
tag blame

## track

add commit  
mv rm  
tag

## branch

checkout branch  
merge rebase

## undo

reset checkout  
revert

## update

pull fetch  
fetchmerge  
apply

## publish

push  
format-patch

## config

config  
remote

## misc

stash  
cherry-pick  
fsck, gc

# Konfiguration

# Konfiguration

```
~/.gitconfig  
.git/config
```

# Konfiguration

```
~/.gitconfig  
.git/config
```

```
git config -global user.name "Federico Hernandez"  
git config -global user.email "ultrafredde@gmail.com"
```

Hands-on



```
Filesystem git clone /foo/bar/repo.git
```

Filesystem `git clone /foo/bar/repo.git`

ssh `git clone username@example.com:repo.git`

```
Filesystem git clone /foo/bar/repo.git  
ssh        git clone username@example.com:repo.git  
git        git clone git://example.com/repo.git
```

```
Filesystem git clone /foo/bar/repo.git
ssh        git clone username@example.com:repo.git
git        git clone git://example.com/repo.git
http       git clone git://example.com/repo.git
```

# Tools

Webserver

Webserver

git server

Webserver

git server: gitosis

Webserver

git server: gitosis

GUI

# Ressourcen

git Homepage <http://git-scm.com>

## Ressourcen

git Homepage <http://git-scm.com>

Progit Buch <http://progit.org>

## Resourcen

git Homepage <http://git-scm.com>

Progit Buch <http://progit.org>

Learn git <http://learn.github.com>

## Resourcen

git Homepage <http://git-scm.com>  
Progit Buch <http://progit.org>  
Learn git <http://learn.github.com>  
Screencast <http://gitcasts.com>

## Resourcen

git Homepage <http://git-scm.com>

Progit Buch <http://progit.org>

Learn git <http://learn.github.com>

Screencast <http://gitcasts.com>

Cheat sheet <http://swxruby.org/git-cheat-sheet.pdf>

## Resourcen

git Homepage <http://git-scm.com>

Progit Buch <http://progit.org>

Learn git <http://learn.github.com>

Screencast <http://gitcasts.com>

Cheat sheet <http://swxruby.org/git-cheat-sheet.pdf>

The git Parabel

## Resourcen

git Homepage <http://git-scm.com>

Progit Buch <http://progit.org>

Learn git <http://learn.github.com>

Screencast <http://gitcasts.com>

Cheat sheet <http://swxruby.org/git-cheat-sheet.pdf>

The git Parabel

Tv's cobweb: Git for Computer Scientists

# Resourcen

git Homepage <http://git-scm.com>

Progit Buch <http://progit.org>

Learn git <http://learn.github.com>

Screencast <http://gitcasts.com>

Cheat sheet <http://swxruby.org/git-cheat-sheet.pdf>

The git Parabel

Tv's cobweb: Git for Computer Scientists

Version Control for Designers

# Resourcen

git Homepage <http://git-scm.com>  
Progit Buch <http://progit.org>  
Learn git <http://learn.github.com>  
Screencast <http://gitcasts.com>  
Cheat sheet <http://swxruby.org/git-cheat-sheet.pdf>  
The git Parabel  
Tv's cobweb: Git for Computer Scientists  
Version Control for Designers

Github <http://github.com>  
Gitorious <http://gitorious.org>

Fragen?

# Fragen?

Federico Hernandez <ultrafredde@gmail.com>  
<http://tasktools.org/talks>