

[illegible]

LG Programmiersysteme
FernUniversität Hagen, Universitätsstraße 11 (IZ), 58097 Hagen

Agenda

- Definition
- Versionsverwaltung
 - Kollaboration
 - Grundlagen
- Git
 - Unterschiede
 - Objektmodell
 - Praxis
- Weiteres zu Git
 - Einschränkungen
 - Statistik - Der Linux Kernel
- Fazit



🔑 **1. git commit**

📁 **2. git push**

🚪 **3. leave building**

¹ <https://github.com/louim/in-case-of-fire>, letzter Zugriff 19.02.2018

Was bedeutet

Reviews

Versionsverwaltung?

Nachvollziehbar

„Version Control: The Freedom to Delete“¹

Eindeutig

Reproduzierbar

Warum?

Kollaboration

¹ Jez Humble und David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010.

zeitliche und geografische
Unabhängigkeit

Warum

Konflikte

Kollaboration?

„Keep Absolutely Everything in Version Control“¹

Teams

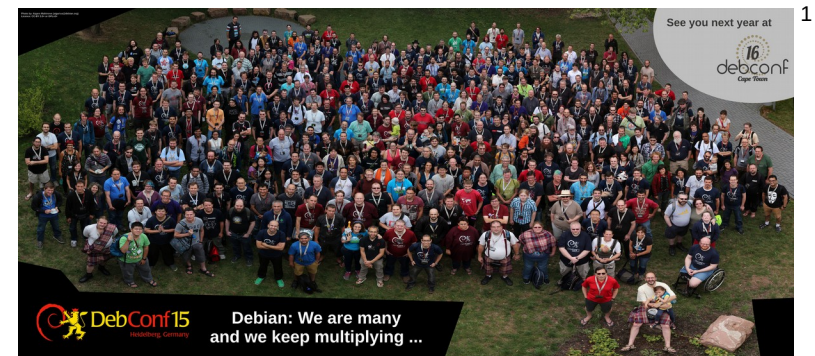
Integration

Asynchron

¹ Jez Humble und David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010.

Förderung der Zusammenarbeit durch

- schnelle Integration von neuen Mitarbeitern,
- Wissensaustausch,
- Nachvollziehbarkeit,
- Aktualität,
- Konsistenz,
- Konfliktsteuerung,
- Integration von Prozessen,
- ...



¹ DebConf e.V., <https://debconf15.debconf.org/news.xhtml>,
letzter Zugriff 19.02.1018

Allgemeine

Repository

HEAD

Grundlagen

Grundbegriffe

Tag

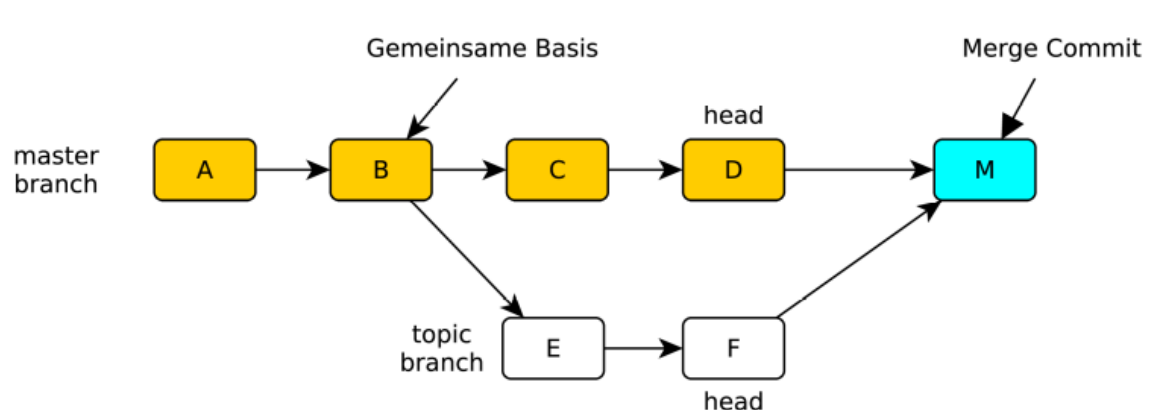
Checkout

Merge

Branch

Grundbegriffe

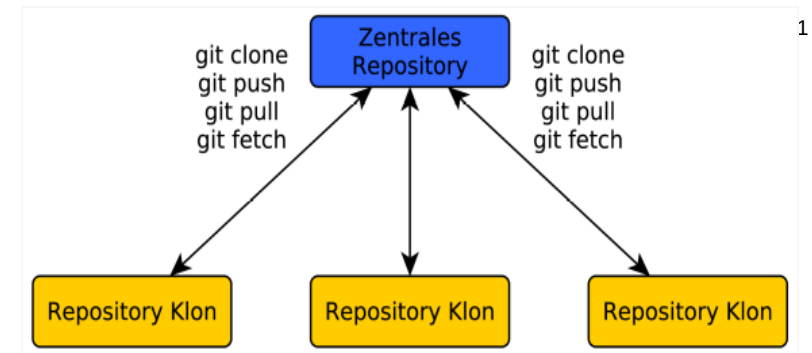
- Repository
- Commit
- Branch
- Tag
- HEAD
- Merge
- Checkout vs Clone



¹ Angelehnt an: Haenel Valentin and Plenz Julius. Git - Verteilte Versionsverwaltung für Code und Dokumente. Open Source Press, München, 2011, S. 83.

Arten von Versionsverwaltungssystemen

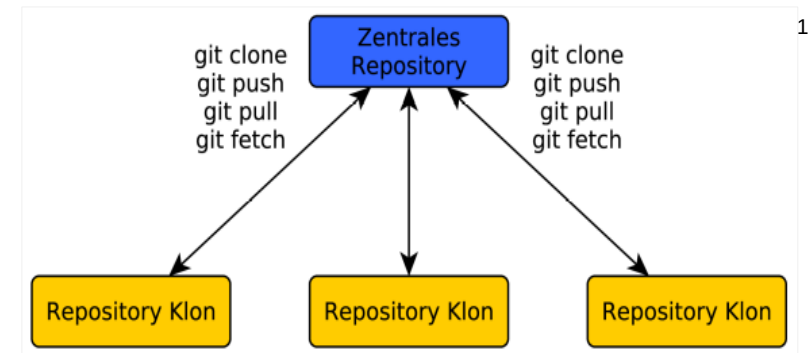
- Lokal
- Zentral
- Verteilt
- Streaming



¹ Angelehnt an: Haenel Valentin and Plenz Julius. Git - Verteilte Versionsverwaltung für Code und Dokumente. Open Source Press, München, 2011, S. 138.

Verteilte Versionskontrollsysteme

- Keine Serverkomponente nötig
- Verschiedene Quellen
- Kein Netzwerk nötig
- Weniger Datenverluste
- Sortierung vor Veröffentlichung



¹ Angelehnt an: Haenel Valentin und Plenz Julius. Git - Verteilte Versionsverwaltung für Code und Dokumente. Open Source Press, München, 2011, S. 138.

Versionskontrolle mit

Index
Stage

HEAD

Git!

„I'm an egoistical bastard, and I name all my projects after myself. First 'Linux', now 'Git'“¹

Objektmodell

Tags
Branches

*The stupid
content tracker*

Trees

SHA-1

¹ Linus Torvalds. Git Faq. <https://git.wiki.kernel.org>,
Linux Kernel Organization, letzter Zugriff 03.11.2017

Struktur

- Trunk vs Master
- Branches und Tags
- Objekt vs Inhalt
- Revision vs SHA-1



1

¹ Kelly Bailey, Wikimedia Commons,
https://commons.wikimedia.org/wiki/File:Two_different_shoes_on.jpg#

A Tale of Three Trees¹

- Arbeitsbereich
- Index
- Repository



¹ Chacon Scott. A tale of three trees. <https://speakerdeck.com/schacon/a-tale-of-three-trees>. Letzter Zugriff 28.11.2017 19:36, 2011.

² Angelehnt an: Haenel Valentin and Plenz Julius. Git - Verteilte Versionsverwaltung für Code und Dokumente. Open Source Press, München, 2011, S. 34.

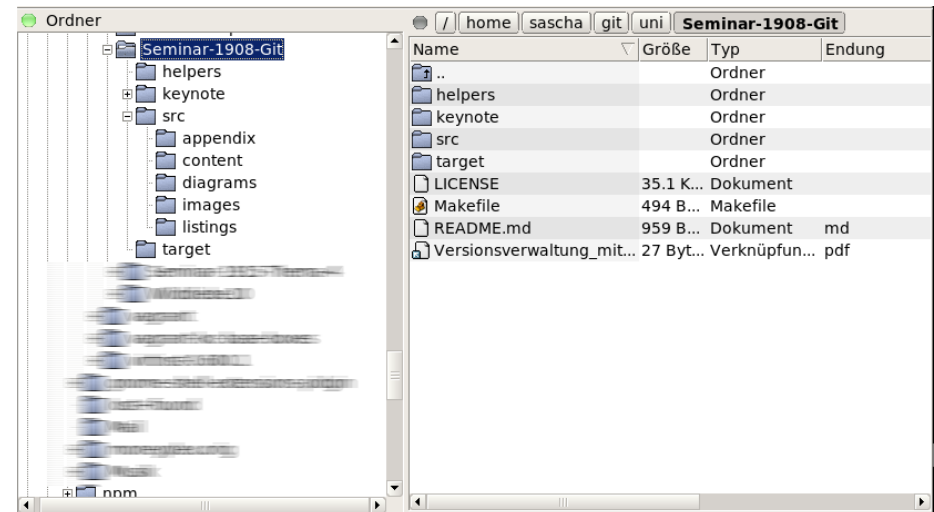
Ein Baum besteht aus

- Dateien
- und Bäumen

```
$ tree
```

```
.  
|-- helpers  
|   |-- git-stats  
|-- LICENSE
```

```
1 directory, 2 files
```



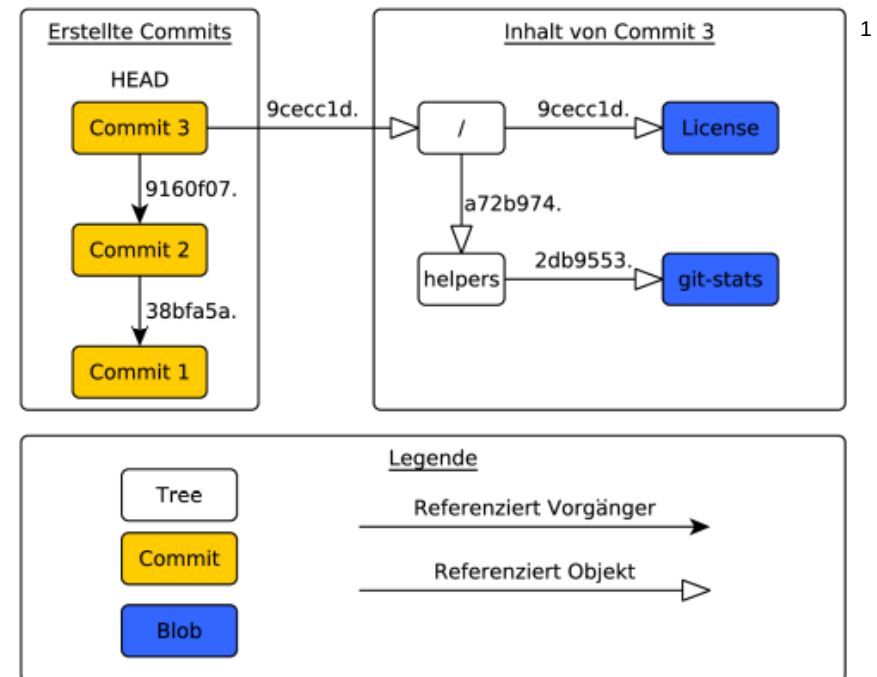
It's all about the trees, baby¹

¹ Chacon Scott. A tale of three trees. <https://speakerdeck.com/schacon/a-tale-of-three-trees>. Letzter Zugriff 28.11.2017 19:36, 2011.

Objekttypen

- Tree
- Blob
- Commit
- Tag
- Branch?

```
$ git ls-tree HEAD
100644 blob 9cecc1d4669ee8af2ca727a5d8cde10cd8b2d7cc LICENSE
040000 tree a72b974ebc96599b9ab0fc82fe8b3457da148744 helpers
```

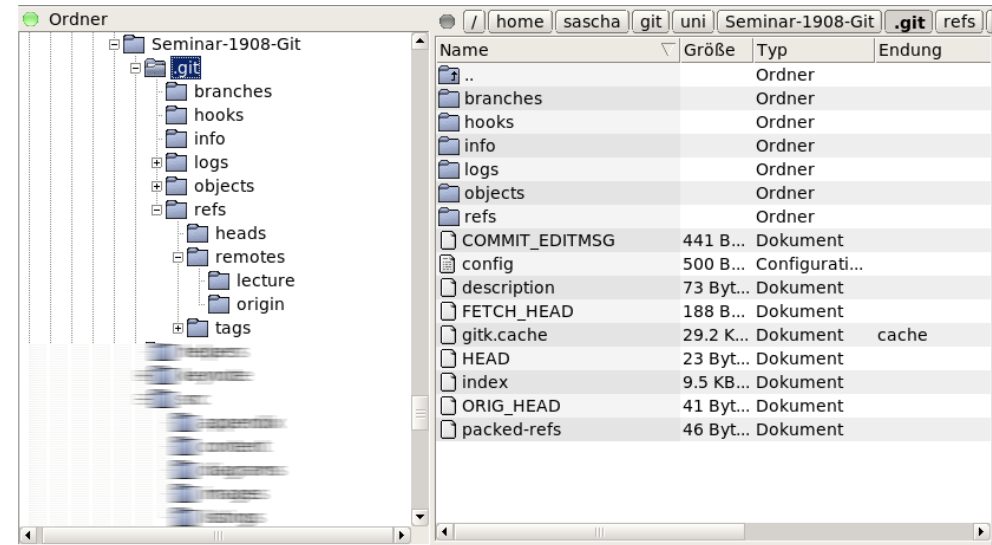


¹ Angelehnt an: Haenel Valentin and Plenz Julius. Git - Verteilte Versionsverwaltung für Code und Dokumente. Open Source Press, München, 2011, S. 53.

Objektmodell

Referenzen

- remotes
- heads
- tags



```
$ cat .git/refs/heads/master  
614195f7f35391a18486f1da885776bc9cbb7f0b
```

```
$ git ls-tree HEAD  
100644 blob 9cecc1d4669ee8af2ca727a5d8cde10cd8b2d7cc LICENSE  
040000 tree a72b974ebc96599b9ab0fc82fe8b3457da148744 helpers
```

Erstellen eines Repositorys

1

- `$ git init <name>`

```
$ git init git-example.git
```

Leeres Git-Repository in `/home/seminar/git-example.git/.git/` initialisiert

- `$ git config <option> <parameter>`

```
$ git config --global user.name "Markus Moeglich"
```

```
$ git config --global user.email markus@moeglich.de
```

```
$ git config --global color.ui auto
```

```
$ git config --global --list
```

```
user.name=Markus Moeglich
```

```
user.email=markus@moeglich.de
```

```
color.ui=auto
```


Dateien hinzufügen

1

- `$ git status`

```
$ git status
```

```
Auf Branch master
```

```
Noch keine Commits
```

```
nichts zu committen (erstellen/kopieren Sie Dateien und benutzen  
Sie "git add" zum Versionieren)
```

- `$ git add <datei>`

```
$ git add LICENSE
```

Dateien hinzufügen

1

- \$ git status

```
$ git status
```

```
Auf Branch master
```

```
Noch keine Commits
```

```
zum Commit vorgemerkte Änderungen:
```

```
(benutzen Sie "git rm --cached <Datei>..." zum Entfernen aus der Staging-  
Area)
```

```
neue Datei:  LICENSE
```

```
Unversionierte Dateien:
```

```
(benutzen Sie "git add <Datei>...", um die Änderungen zum Commit  
vorzumerken)
```

```
helpers/
```

Dateien hinzufügen

1

- \$ git commit [options]

```
$ git commit
[master 86c3661] Add gpl license file
1 file changed, 1674 insertions(+)
```

- \$ git log [options]

```
$ git log --oneline
878b171 Add executable bit to helper script.
ce31bd5 Add helper script to generate author statistics.
86c3661 Add gpl license file
```

Entfernte Repositories

- `$ git clone <uri>`

```
$ git clone https://github.com/sagiru/git-example.git
Klone nach 'git-example' ...
remote: Counting objects: 10, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 10 (delta 0), reused 10 (delta 0), pack-reused 0
Entpacke Objekte: 100% (10/10), Fertig.
```

- `$ git push`

```
$ git push
Zähle Objekte: 3, Fertig.
Delta compression using up to 4 threads.
Komprimiere Objekte: 100% (2/2), Fertig.
Schreibe Objekte: 100% (3/3), 355 bytes | 355.00 KiB/s, Fertig.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:sagiru/git-example.git
ce31bd5..878b171 master -> master
```

Weitere Befehle

- `$ git branch <name> [id]`
- `$ git tag <name> [typ] [id]`
- `$ git show [options]`
- `$ git log [options]`
- `$ git fetch [options]`
- `$ git merge [options]`
- `$ man git / $ git <command> --help`

Berechtigungen

Zahlen, Fakten und Einschränkungen zu

Alles oder Nichts

Git

Goddamn idiotic truckload of sh*t: when it breaks¹

Binärdateien

Lernkurve

Werkzeuge

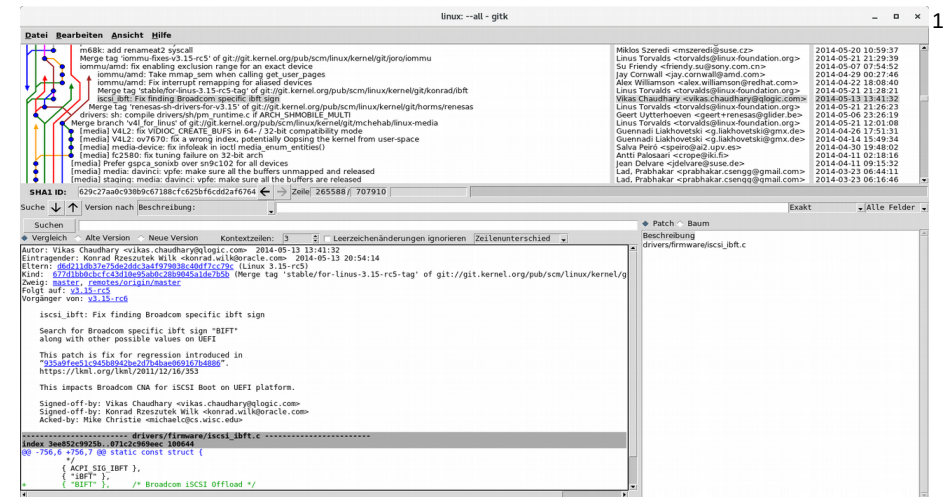
Kopflos?

¹ Linus Torvalds. Git Faq. <https://git.wiki.kernel.org>,
Linux Kernel Organization, letzter Zugriff 03.11.2017

Weiteres zu Git

Einschränkungen

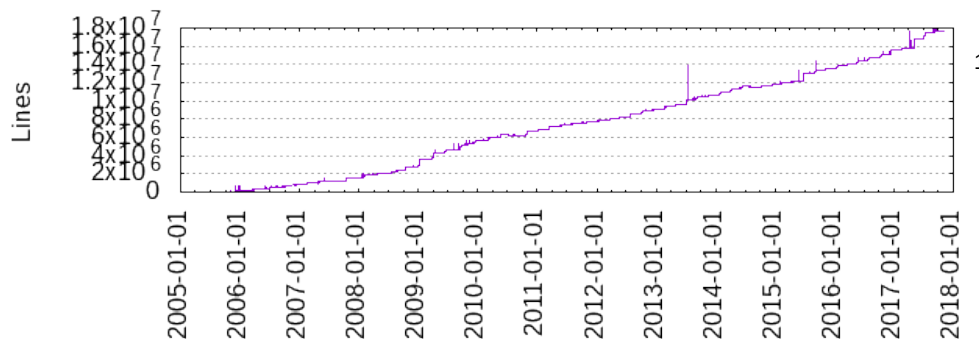
- Binärdateien
- Alles oder nichts
- Grafische Werkzeuge



¹<https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git>,
Stand 13.11.2017

Statistik - Der Linux Kernel

- ~18M Zeilen Code
- ~61T Dateien
- ~664T Commits
- ~17T Autoren



```
commit 1da177e4c3f41524e886b7f1b8a0c1fc7321cac2
Author: Linus Torvalds <torvalds@ppc970.osdl.org>
Date: Sat Apr 16 15:20:36 2005 -0700
```

1

Linux-2.6.12-rc2

Initial git repository build. I'm not bothering with the full history, even though we have it. We can create a separate "historical" git archive of that later if we want to, and in the meantime it's about 3.2GB when imported into git - space that would just make the early git days unnecessarily complicated, when we don't have a lot of good infrastructure for it.

Let it rip!

¹<https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git>,
Stand 13.11.2017

Git

Version Control: The Freedom to Delete¹

¹ Jez Humble and David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010.