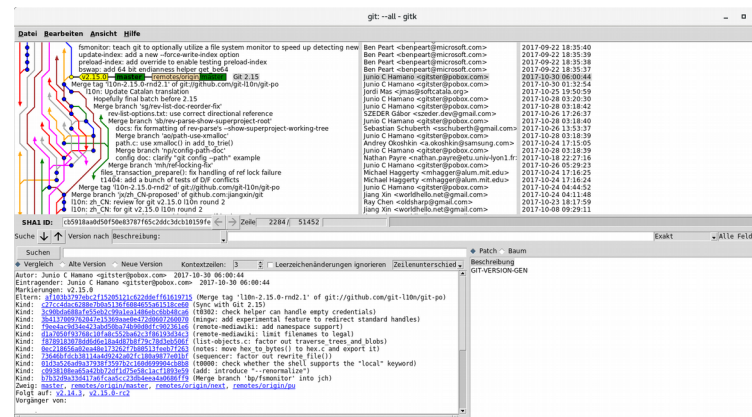


Versionsverwaltung mit Git



Seminararbeit im Studiengang Bachelor Informatik

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

Agenda

- I. Einleitung
- II. Versionsverwaltung
 - Kollaboration
 - Grundlagen
- III. Git
 - Unterschiede
 - Objektmodell
- IV. Weiteres zu Git
 - Einschränkungen
 - Statistik - Der Linux Kernel
- V. Fazit

In case of fire



🔑 **1. git commit**

📁 **2. git push**

🚶 **3. leave building**

¹ <https://github.com/louim/in-case-of-fire>

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

Unterstützung der

Konflikte

Kollaboration?

„Keep Absolutely Everything in Version Control“¹

Teams

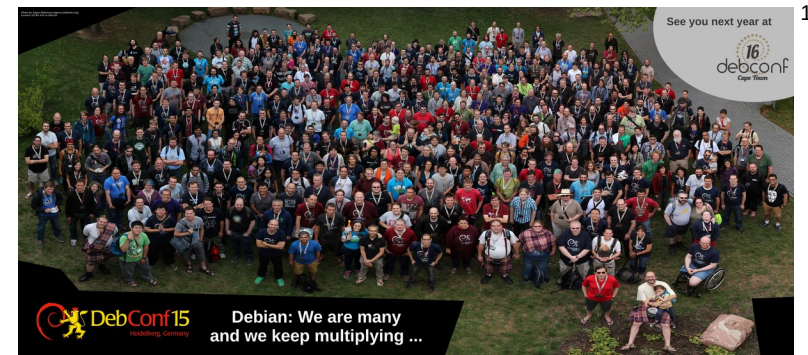
Integration

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

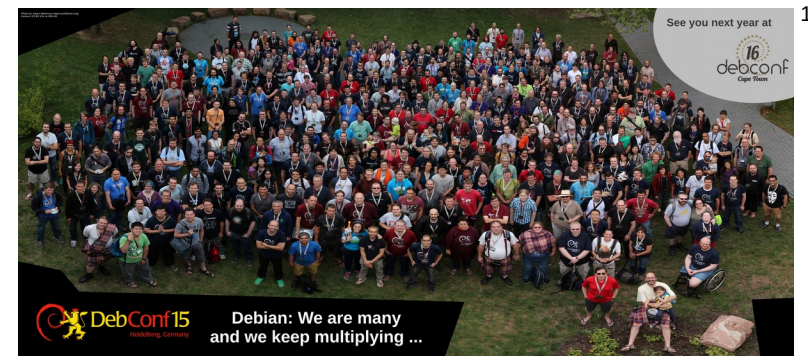
- schnelle Integration von neuen Mitarbeitern,



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

Förderung der Zusammenarbeit durch

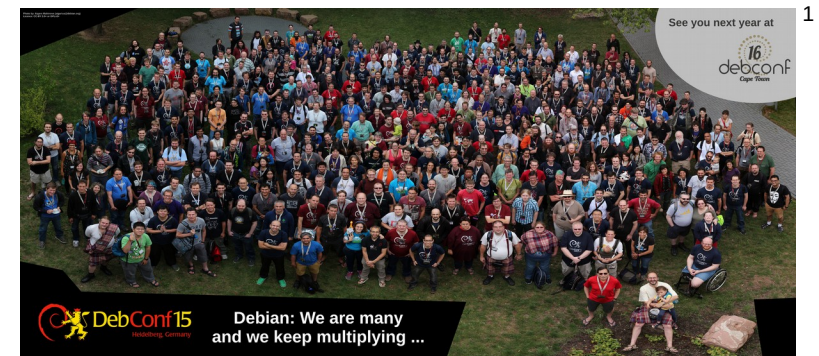
- schnelle Integration von neuen Mitarbeitern,
- Integration von Prozessen,



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

Förderung der Zusammenarbeit durch

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



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

Allgemeine

Repository

HEAD

Grundlagen

Grundbegriffe

Tag

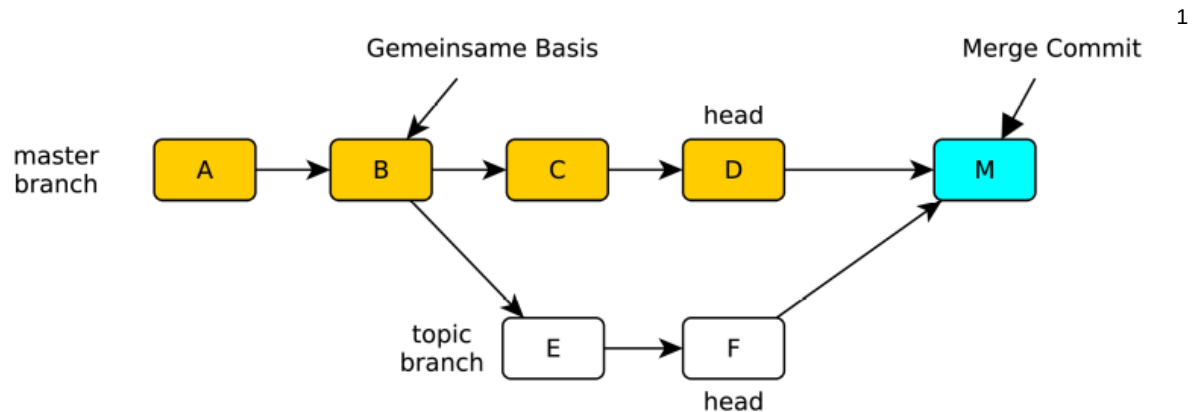
Checkout

Merge

Branch

Grundbegriffe

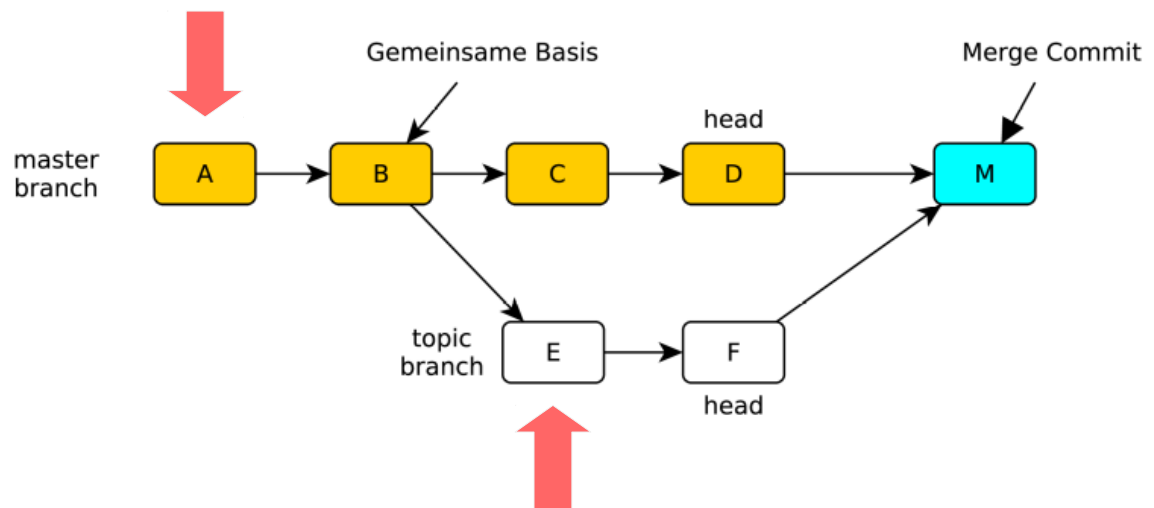
- Repository



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

Grundbegriffe

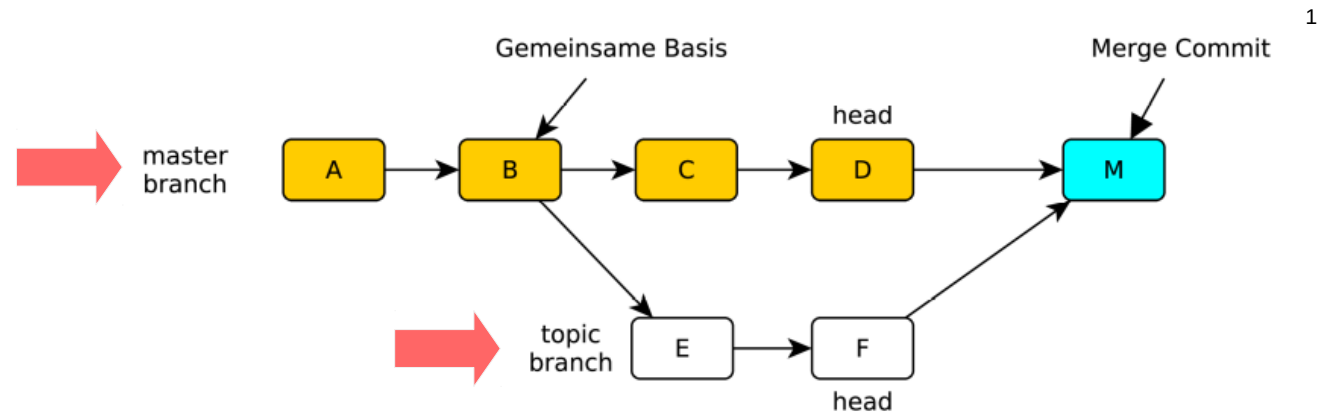
- Repository
- Commit



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

Grundbegriffe

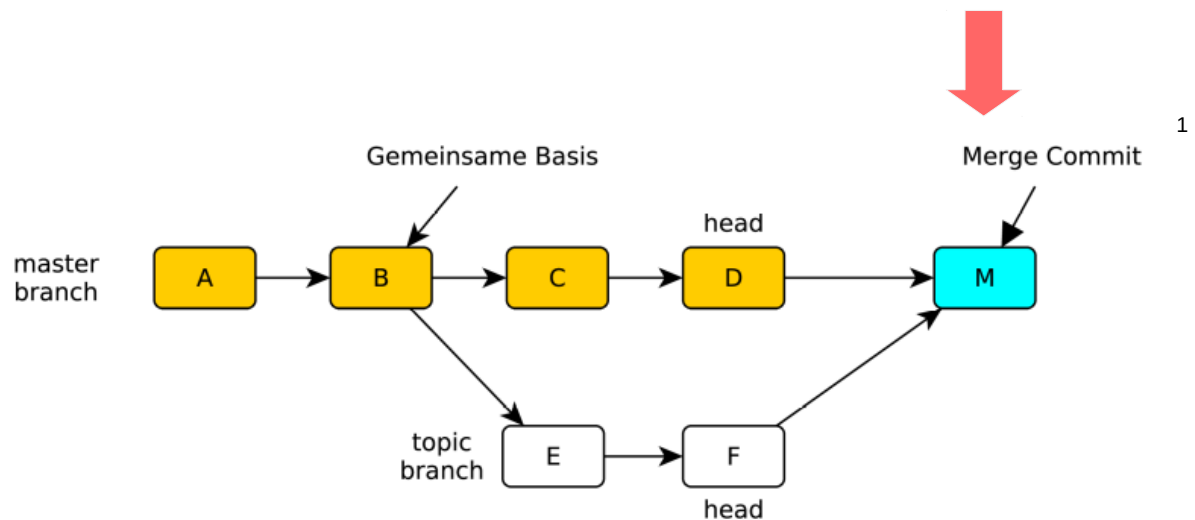
- Repository
- Commit
- Branch



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

Grundbegriffe

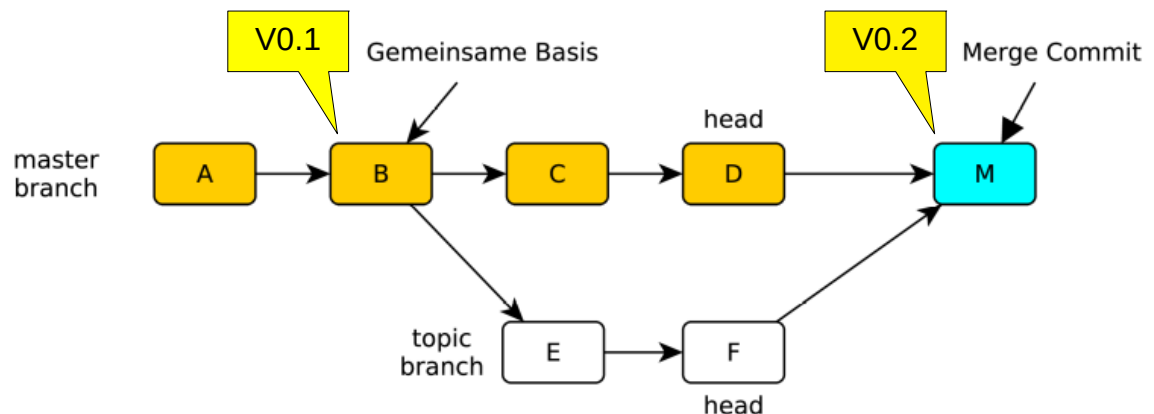
- Repository
- Commit
- Branch
- Merge



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

Grundbegriffe

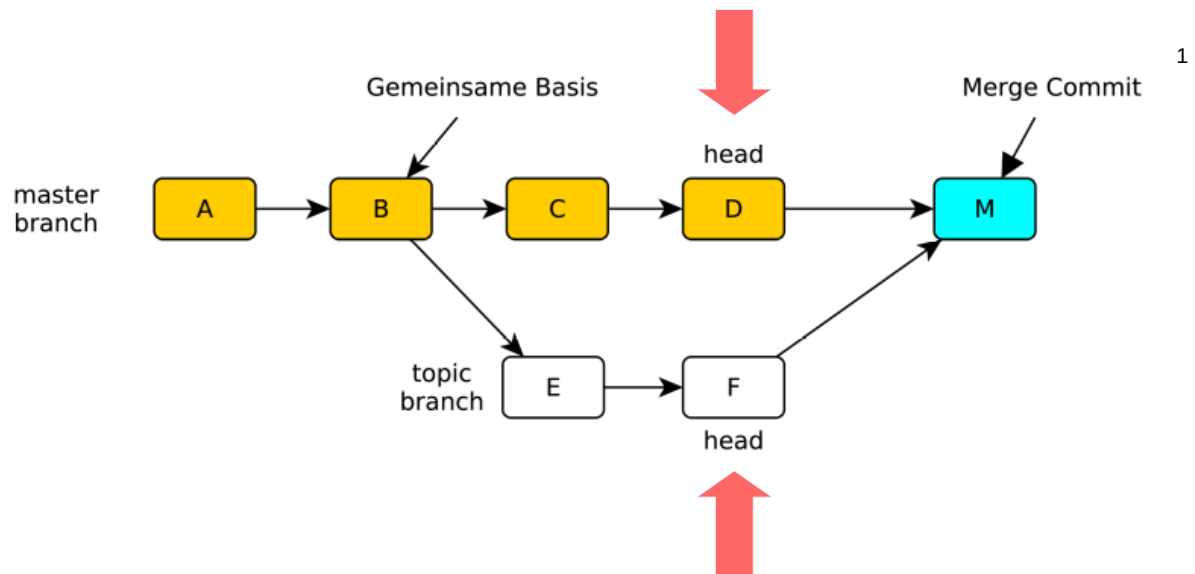
- Repository
- Commit
- Branch
- Merge
- Tag



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

Grundbegriffe

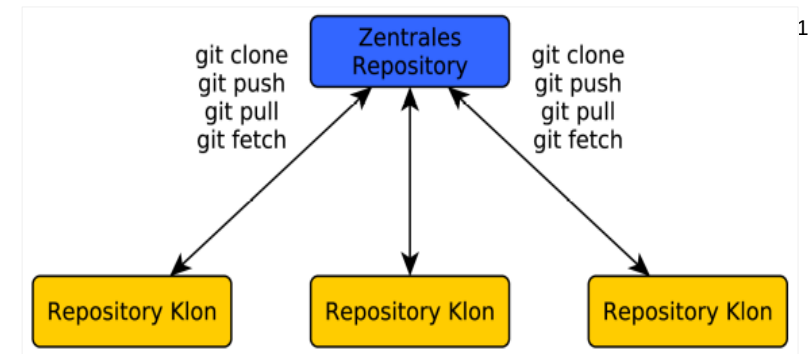
- Repository
- Commit
- Branch
- Merge
- Tag
- HEAD



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

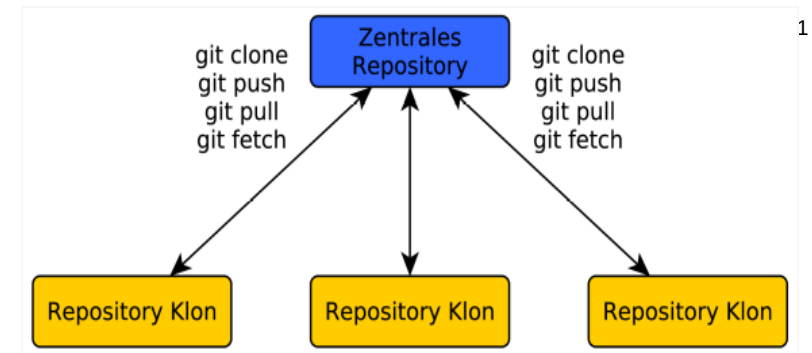
- Lokal



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

Arten von Versionsverwaltungssystemen

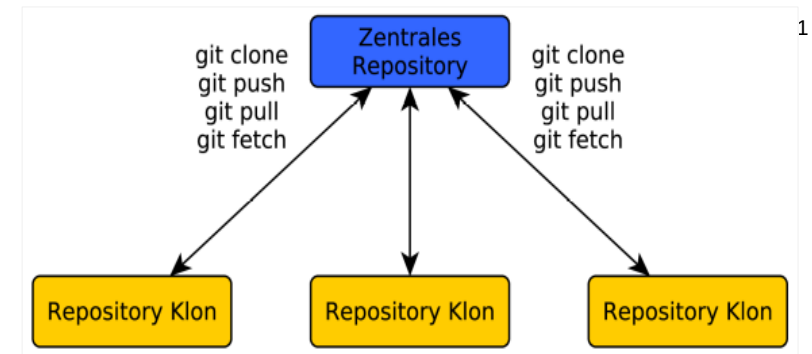
- Lokal
- Zentral



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

Arten von Versionsverwaltungssystemen

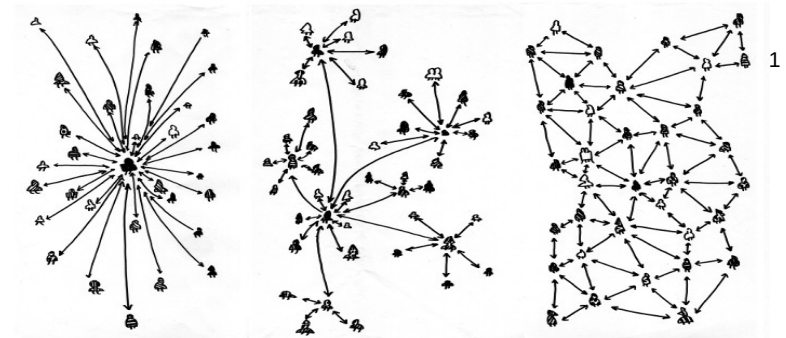
- Lokal
- Zentral
- Verteilt



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

Verteilte Versionskontrollsysteme

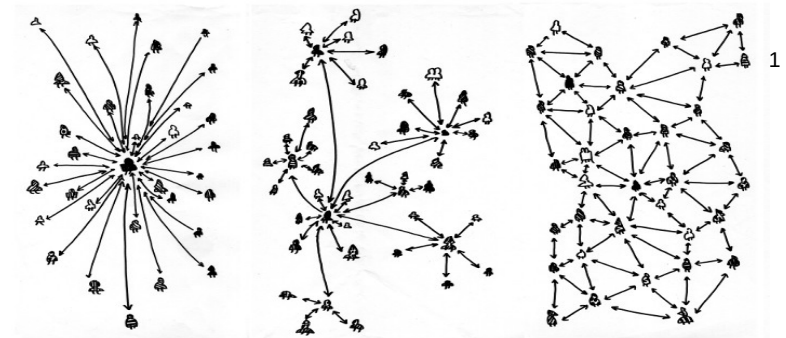
- Keine Serverkomponente nötig



¹ Kelly Bailey, Wikimedia Commons,
https://commons.wikimedia.org/wiki/File:P2P_Topology.jpg#

Verteilte Versionskontrollsysteme

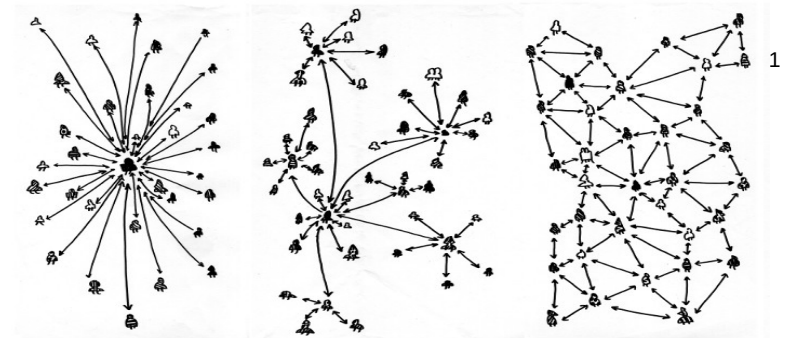
- Keine Serverkomponente nötig
- Verschiedene Quellen



¹ Kelly Bailey, Wikimedia Commons,
https://commons.wikimedia.org/wiki/File:P2P_Topology.jpg#

Verteilte Versionskontrollsysteme

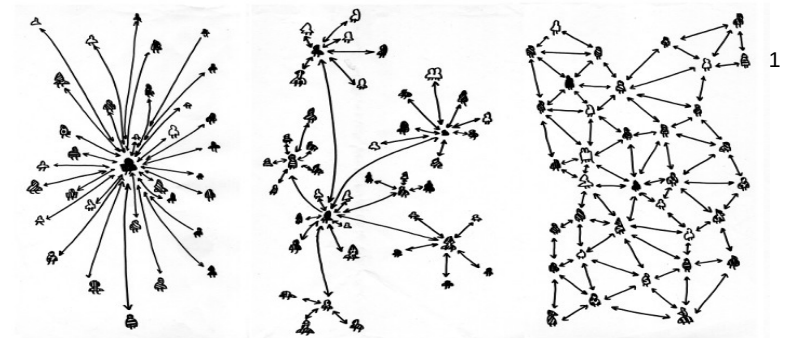
- Keine Serverkomponente nötig
- Verschiedene Quellen
- Weniger Datenverluste



¹ Kelly Bailey, Wikimedia Commons,
https://commons.wikimedia.org/wiki/File:P2P_Topology.jpg#

Verteilte Versionskontrollsysteme

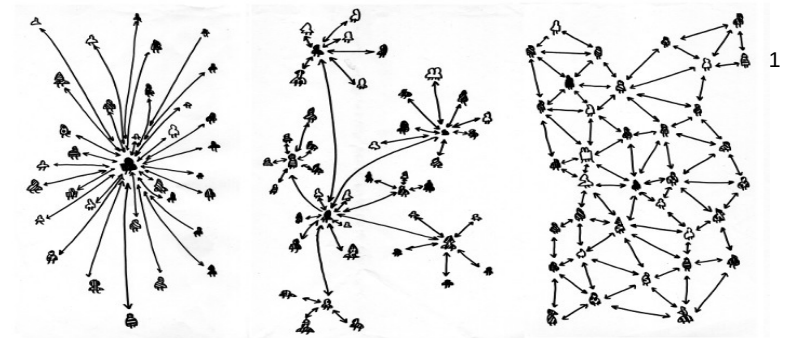
- Keine Serverkomponente nötig
- Verschiedene Quellen
- Weniger Datenverluste
- Kein Netzwerk nötig



¹ Kelly Bailey, Wikimedia Commons,
https://commons.wikimedia.org/wiki/File:P2P_Topology.jpg#

Verteilte Versionskontrollsysteme

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



¹ Kelly Bailey, Wikimedia Commons,
https://commons.wikimedia.org/wiki/File:P2P_Topology.jpg#

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



1

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

Struktur

- Trunk vs Master
- Branches und Tags



1

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

Struktur

- Trunk vs Master
- Branches und Tags
- Objekt vs Inhalt



1

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

Struktur

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



1

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

Struktur

- Trunk vs Master
- Branches und Tags
- Objekt vs Inhalt
- Revision vs SHA
- Bäume?



1

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

A Tale of Three Trees¹

- Arbeitsbereich



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.

A Tale of Three Trees¹

- Arbeitsbereich
- Repository

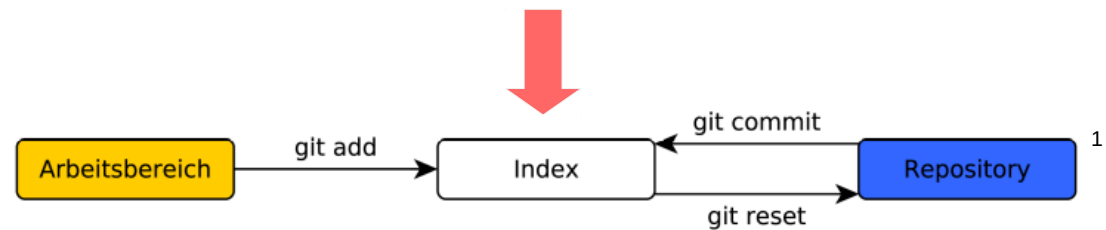


¹ Chacon Scott. A tale of three trees. <https://speakerdeck.com/schacon/a-tale-of-three-trees>. Letzter Zugriff 28.11.2017 19:36, 2011.

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

A Tale of Three Trees¹

- Arbeitsbereich
- Repository
- Index / Stage



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.

Objektmodell

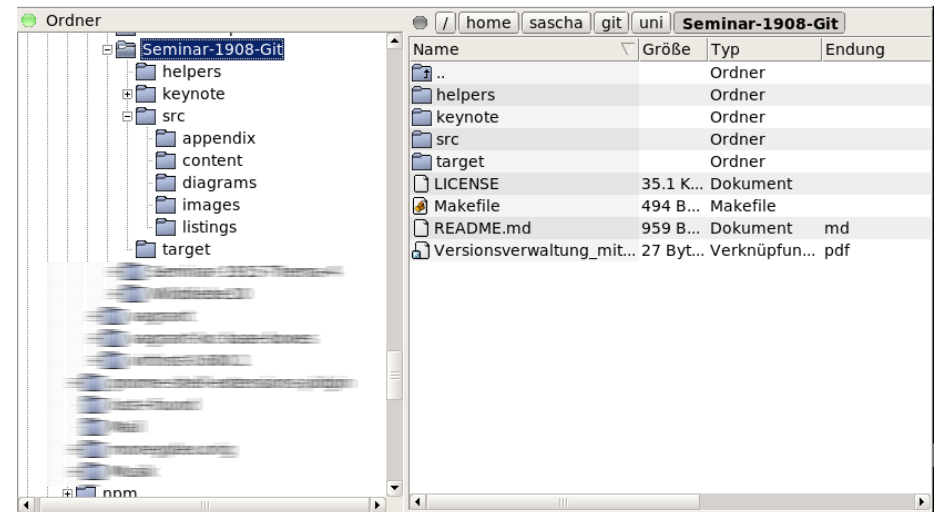
Ein Baum besteht aus

- Dateien

```
$ tree
```

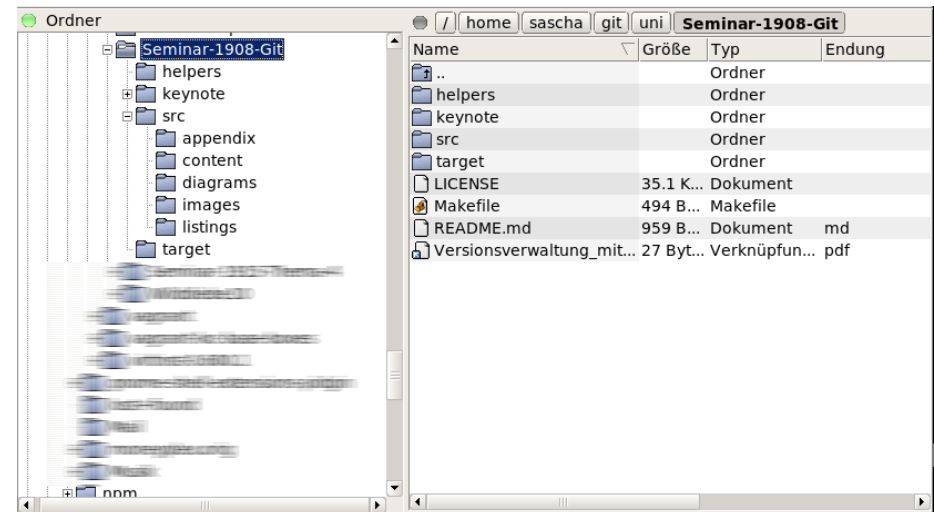
```
.  
|-- helpers  
|   |-- git-stats  
|-- LICENSE
```

```
1 directory, 2 files
```



Ein Baum besteht aus

- Dateien
- und Bäumen



```
$ tree
```

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

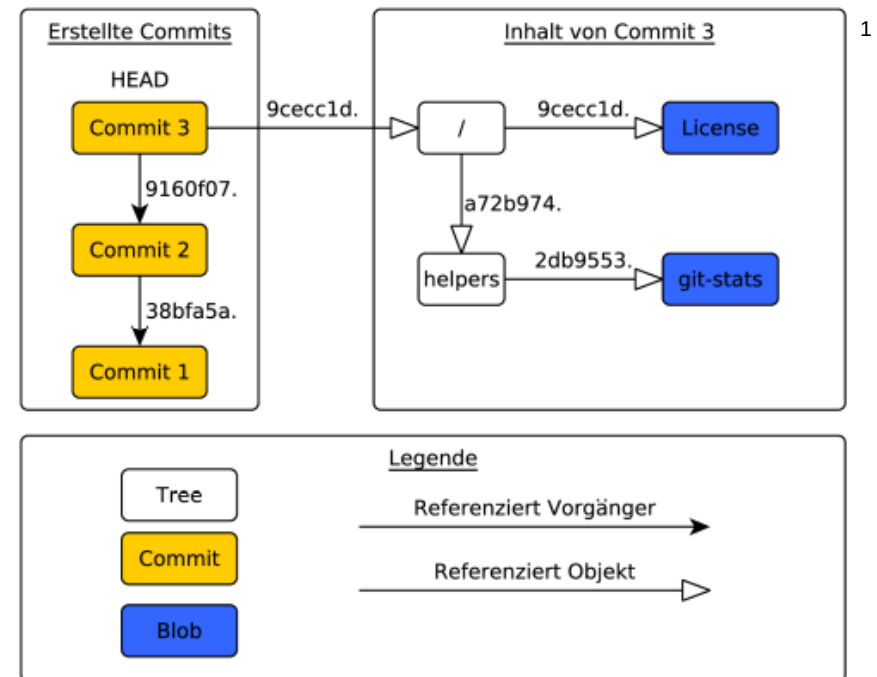
```
1 directory, 2 files
```

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



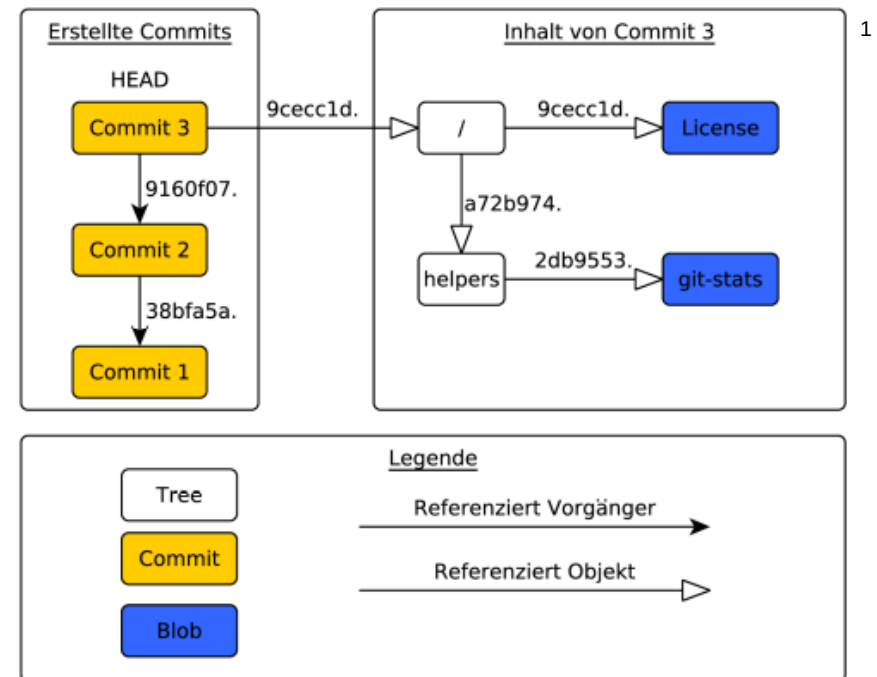
```
$ git ls-tree HEAD
100644 blob 9cecc1d4669ee8af2ca727a5d8cde10cd8b2d7cc LICENSE
040000 tree a72b974ebc96599b9ab0fc82fe8b3457da148744 helpers
```

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

Objekttypen

- Tree
- Blob

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

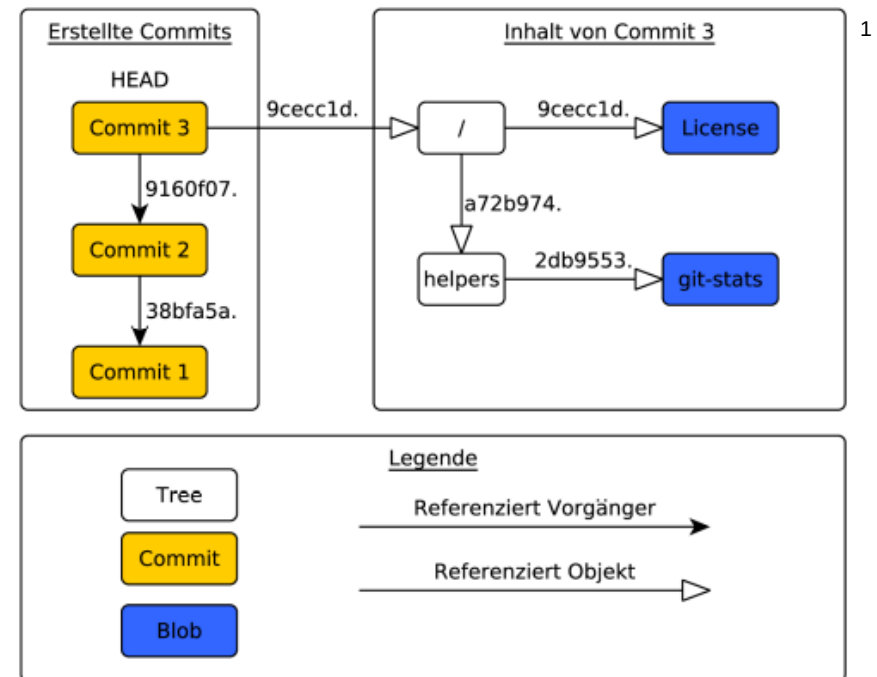


¹ Angelehnt an: Haenel Valentin and Plenz Julius. Git - Verteilte Versionsverwaltung für Code und Dokumente. Open Source Press, München, 2011, S. 53.

Objekttypen

- Tree
- Blob
- Commit

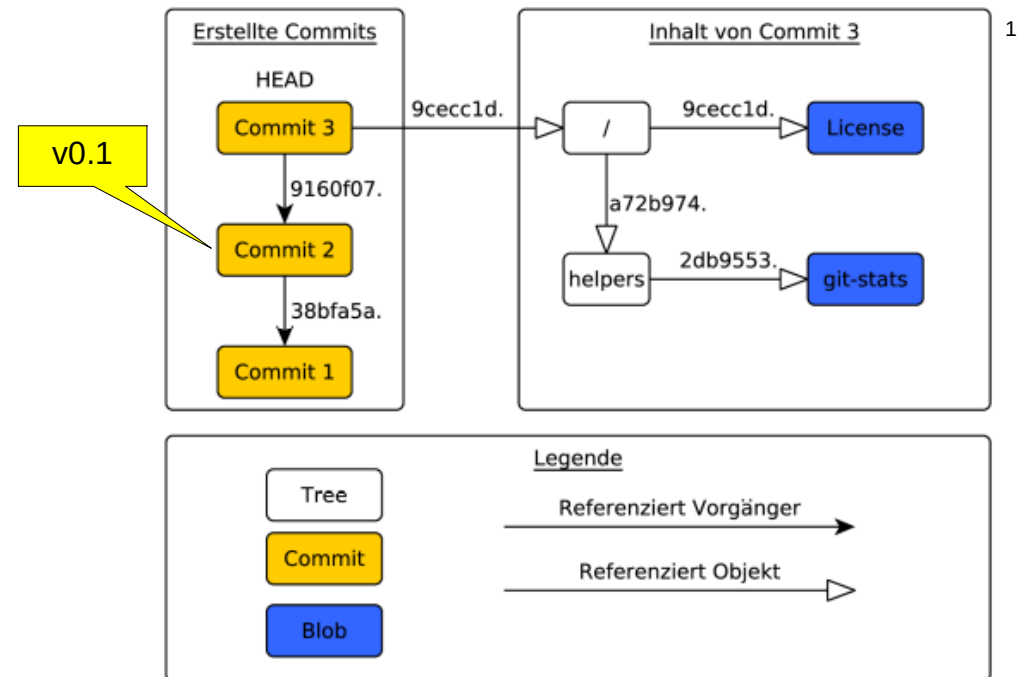
```
$ git ls-tree HEAD
100644 blob 9cecc1d4669ee8af2ca727a5d8cde10cd8b2d7cc LICENSE
040000 tree a72b974ebc96599b9ab0fc82fe8b3457da148744 helpers
```



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

Objekttypen

- Tree
- Blob
- Commit
- Tag



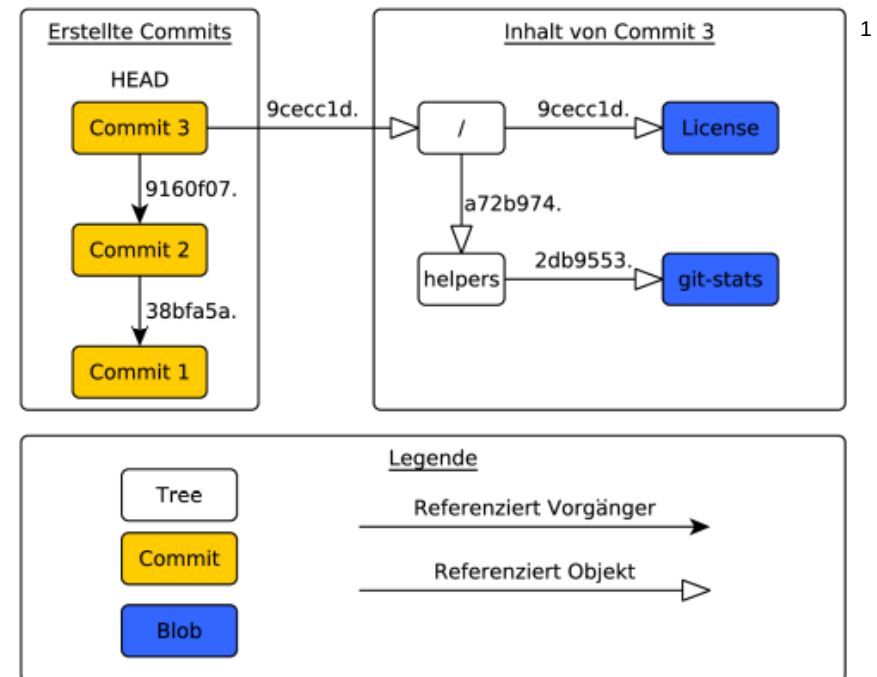
```
$ git ls-tree HEAD
100644 blob 9cecc1d4669ee8af2ca727a5d8cde10cd8b2d7cc LICENSE
040000 tree a72b974ebc96599b9ab0fc82fe8b3457da148744 helpers
```

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

Objekttypen

- Tree
- Blob
- Commit
- Tag
- Branch?

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

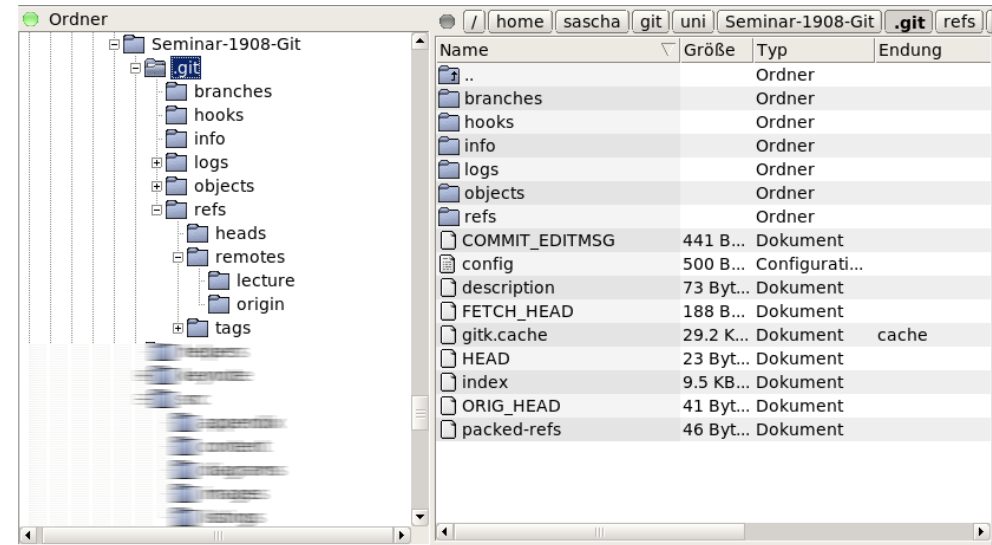


¹ Angelehnt an: Haenel Valentin and Plenz Julius. Git - Verteilte Versionsverwaltung für Code und Dokumente. Open Source Press, München, 2011, S. 53.

Objektmodell

Referenzen

- remotes
- heads



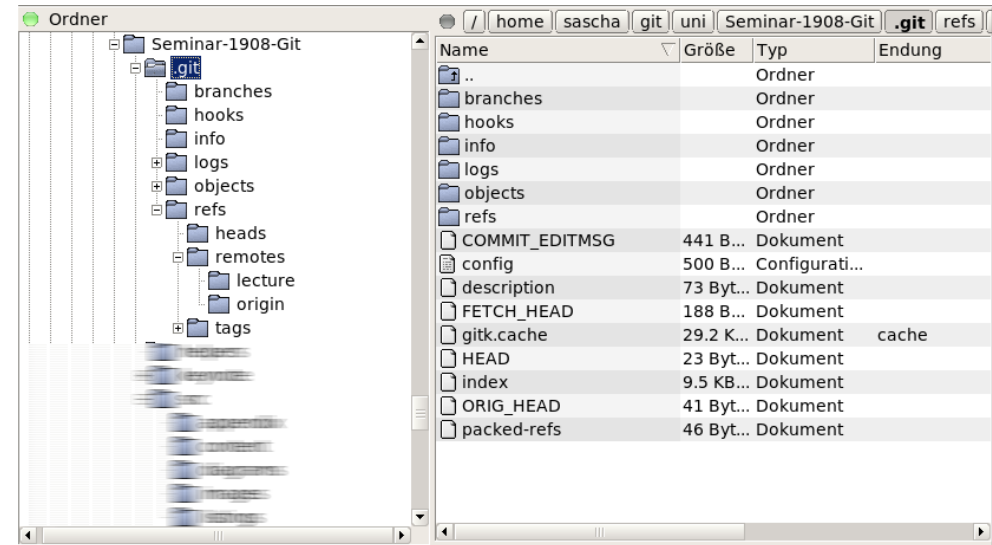
```
$ cat .git/refs/heads/master  
614195f7f35391a18486f1da885776bc9cbb7f0b
```

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

Objektmodell

Referenzen

- remotes
- heads
- tags



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

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


Berechtigungen

Zahlen, Fakten und Einschränkungen zu

Alles oder Nichts

Git

As Torvalds said: „There is no way to do CVS right“¹

Binärdateien

Lernkurve

Werkzeuge

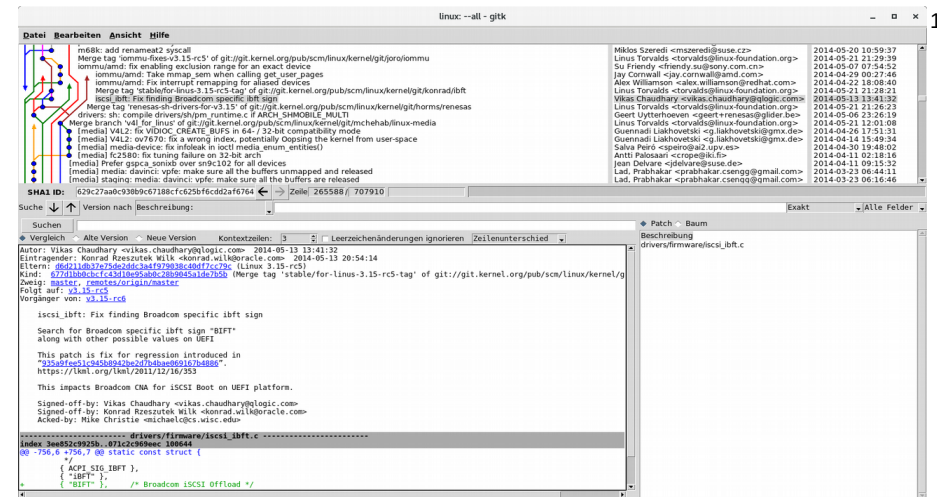
Kopflos?

¹ Humble Jez and Farley David. Continuous Delivery - Reliable Software Releases Through Build, Test and Deployment Automation. Addison Wesley, Boston, 2010, S. 385.

Weiteres zu Git

Einschränkungen

- Grafische Werkzeuge

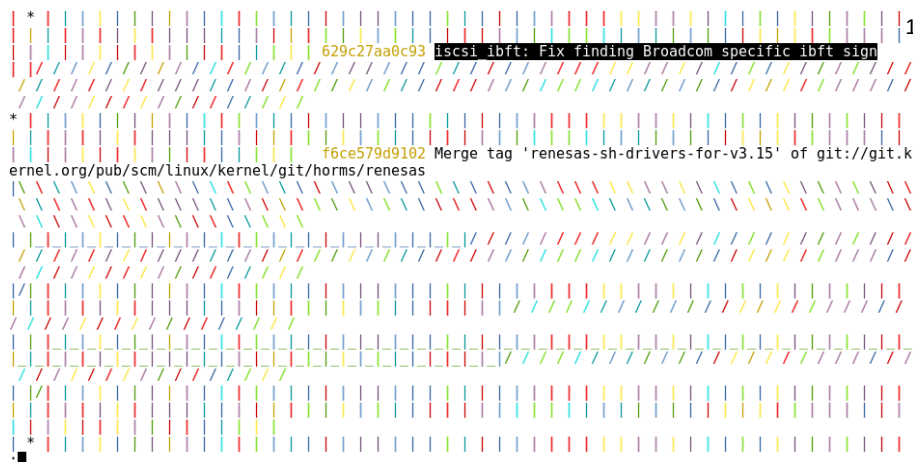
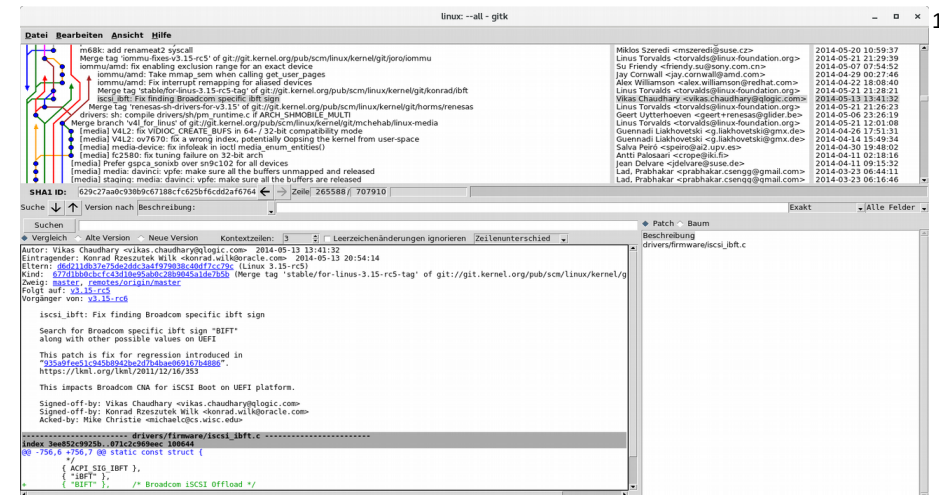


¹<https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git>,
Stand 13.11.2017

Weiteres zu Git

Einschränkungen

- Grafische Werkzeuge
- Binärdateien

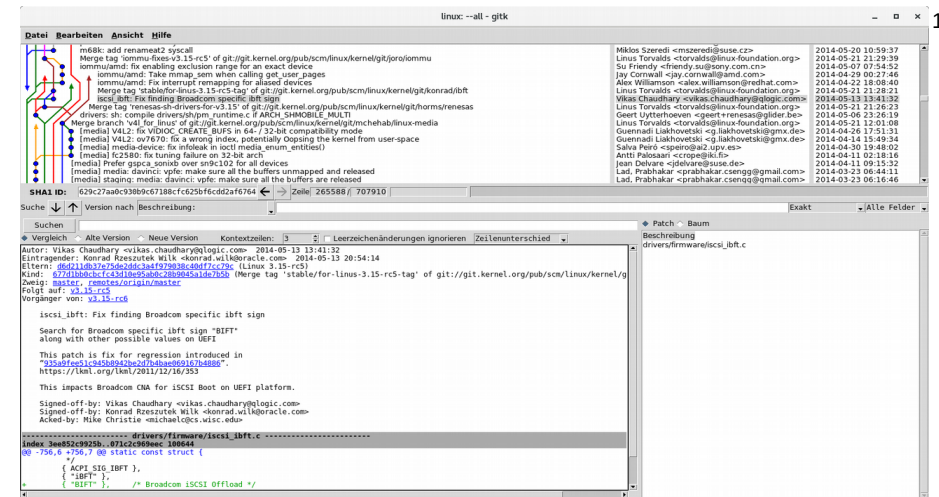


¹<https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git>,
Stand 13.11.2017

Weiteres zu Git

Einschränkungen

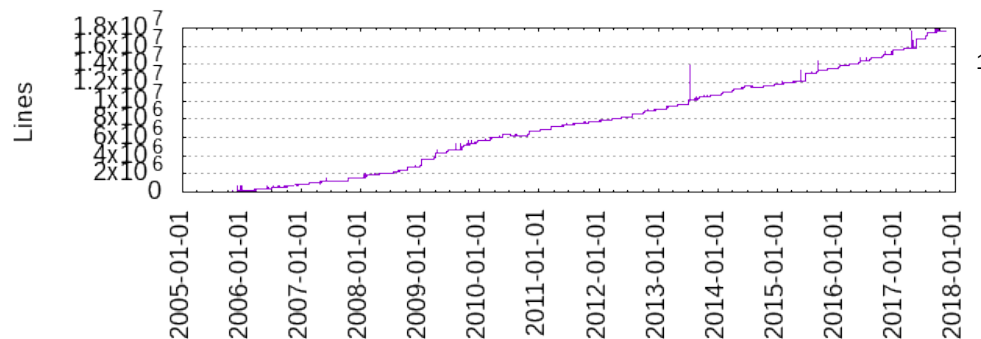
- Grafische Werkzeuge
- Binärdateien
- Alles oder nichts



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

Statistik - Der Linux Kernel

- ~18M Zeilen Code



```
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

Statistik - Der Linux Kernel

- ~18M Zeilen Code
- ~61T Dateien

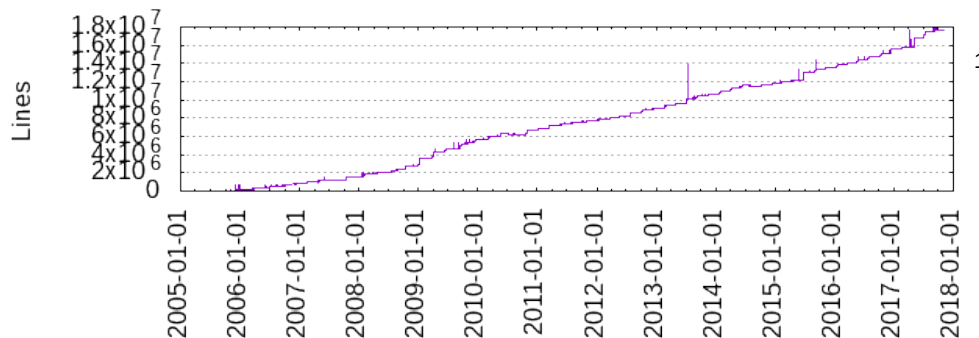
```
commit 1da177e4c3f41524e886b7f1b8a0c1fc7321cac2
Author: Linus Torvalds <torvalds@ppc970.osdl.org>
Date: Sat Apr 16 15:20:36 2005 -0700
```

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!

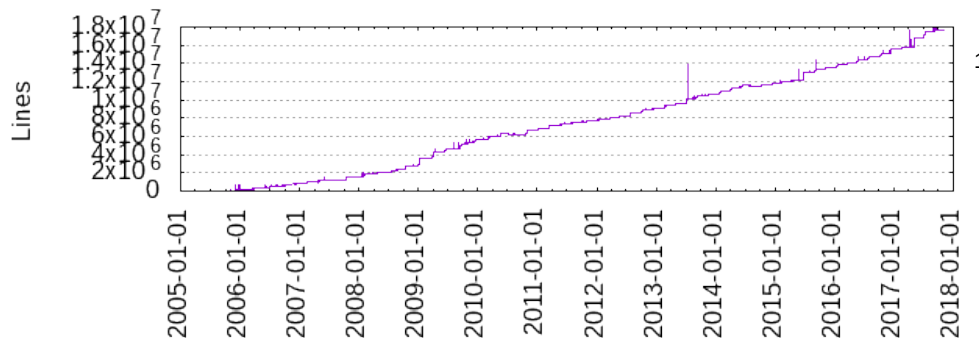


1

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



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

1

Linux-2.6.12-rc2

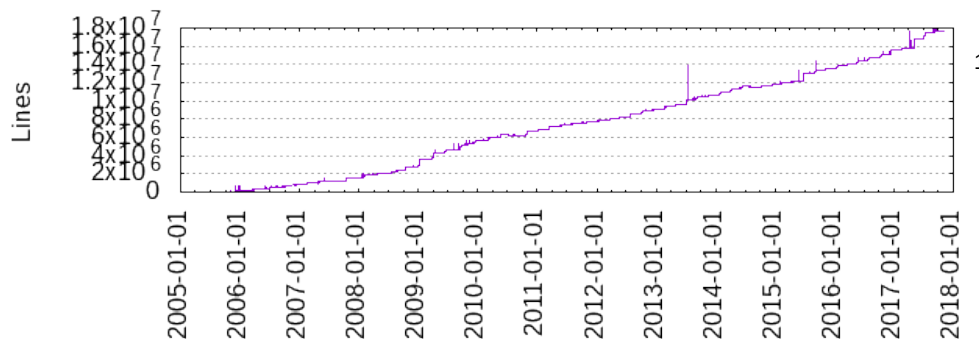
Initial git repository build. I'm not bothering with the full history, even though we have it. We can create a separate "historical" git archive of that later if we want to, and in the meantime it's about 3.2GB when imported into git - space that would just make the early git days unnecessarily complicated,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

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

Trust in Version Control!