Git-Workshop für Einsteiger

Sujeevan Vijayakumaran
zudʒi:væn̩ vɪdʒæjekumaræn
oder auch: Er, dessen Name nicht genannt wird.

Ubucon, Katlenburg

18. Oktober 2014

Inhaltsverzeichnis

Über mich

- Name: Sujeevan Vijayakumaran
- aus: Castrop-Rauxel
- Nickname: svij
- Dualer Bachelor-Student in Dortmund
- 3 Jahre Git-Erfahrung
- Teammitglied und Projektleitung ubuntuusers.de

Was ist eine Versionsverwaltung?

- Verwaltung von Versionen
- vortrag_v1, vortrag_v2, vortrag_v3...
- vortrag_final
- vortrag_final_new

Was kann versioniert werden?

- (Quell-)Texte
- Bilder
- Musik
- u.s.w

→ „Alles!“

Warum eine Versionsverwaltung?

- Nachvollziehbarkeit
- „Springen“ zwischen Versionen
- Teamwork

Angereichert mit weiteren Informationen:

- Autor
- Uhrzeit und Datum
- Änderungsnotiz

Drei Arten der Versionsverwaltung

1. Lokal
2. Zentral
3. Verteilt

Lokale Versionsverwaltung

- lediglich lokal
- häufig nur einzelne Dateien
- nur einzelne Person
- keine Datensicherheit
- SCCS, RCS

Zentrale Versionsverwaltung

- CVS, Subversion (SVN)
- Repository auf zentralem Server
- Arbeitskopie auf Clients
- jede Aktion am Repository nur online

Verteilte Versionsverwaltung

- Git, Bazaar (bzr), Mercurial (hg)
- kompletter Klon liegt auf Clients + Server
- kein unnötiger Netzwerk-Traffic
- höhere Datensicherheit

Geschichtliches

- Entwickler: Linus Torvalds
- 2005 für den Linux-Kernel
- git == Blödmann

Git installieren

Ubuntu/Debian:

```
# apt-get install git
```

Windows:
msysgit: http://git-scm.com/download/win

Repository initialisieren

Projekt-Ordner anlegen:

```
$ mkdir Webseite-mit-Git
```

```
$ cd Webseite-mit-Git
```

Git-Repository initialisieren:

```
$ git init
Initialisierte leeres Git-Repository in
/home/sujee/Webseite-mit-Git/.git/
```

Repository initialisieren

```
$ ls -l .git
insgesamt 32
drwxr-xr-x 2 sujee sujee 4096 20. Jul 16:41 branches
-rw-r--r-- 1 sujee sujee   92 20. Jul 16:41 config
-rw-r--r-- 1 sujee sujee   73 20. Jul 16:41 description
-rw-r--r-- 1 sujee sujee   23 20. Jul 16:41 HEAD
drwxr-xr-x 2 sujee sujee 4096 20. Jul 16:41 hooks
drwxr-xr-x 2 sujee sujee 4096 20. Jul 16:41 info
drwxr-xr-x 4 sujee sujee 4096 20. Jul 16:41 objects
drwxr-xr-x 4 sujee sujee 4096 20. Jul 16:41 refs
```

Git konfigurieren

```
$ git config (--global) user.name "Sujeevan Vijayakumaran"
$ git config (--global) user.email mail@svij.org

$ cat ~/.gitconfig
[user]
        name = Sujeevan Vijayakumaran
        email = mail@svij.org
```

git status

```
$ git status
Auf Branch master

Initialer Commit

nichts zu committen (Erstellen/Kopieren Sie Dateien
und benutzen Sie "git add" zum Beobachten)
```

Vorbereitungen für das Projekt

```
$ wget https://github.com/twbs/bootstrap/releases/
download/v3.2.0/bootstrap-3.2.0-dist.zip
$ unzip bootstrap-3.2.0-dist.zip
$ mv bootstrap-3.2.0-dist/* .
$ rmdir bootstrap-3.2.0-dist
$ rm bootstrap-3.2.0-dist.zip
```

git status

```
$ git status
Auf Branch master

Initialer Commit

Unbeobachtete Dateien:
  (benutzen Sie "git add <Datei>..." um die Änderungen
  zum Commit vorzumerken)

        css/
        fonts/
        js/

nichts zum Commit vorgemerkt, aber es gibt unbeobachtete
Dateien (benutzen Sie "git add" zum Beobachten)
```

git add

```
$ git add css/

Auf Branch master

Initialer Commit

zum Commit vorgemerkte Änderungen:
  (benutzen Sie "git rm --cached <Datei>..." zum Entfernen
  aus der Staging-Area)

  neue Datei:     css/bootstrap-theme.css
  neue Datei:     css/bootstrap-theme.css.map
  neue Datei:     css/bootstrap-theme.min.css
  neue Datei:     css/bootstrap.css
  neue Datei:     css/bootstrap.css.map
  neue Datei:     css/bootstrap.min.css

Unbeobachtete Dateien:
  (benutzen Sie "git add <Datei>..." um die Änderungen
  zum Commit vorzumerken)

  fonts/
  js/
```

git add

```
$ git status
Auf Branch master

Initialer Commit

zum Commit vorgemerkte Änderungen:
  (benutzen Sie "git rm --cached <Datei>..." zum Entfernen
  aus der Staging-Area)

        neue Datei:     css/bootstrap-theme.css
        neue Datei:     css/bootstrap-theme.css.map
        neue Datei:     css/bootstrap-theme.min.css
        neue Datei:     css/bootstrap.css
        neue Datei:     css/bootstrap.css.map
        neue Datei:     css/bootstrap.min.css
        neue Datei:     fonts/glyphicons-halflings-regular.eot

Unbeobachtete Dateien:
  (benutzen Sie "git add <Datei>..." um die Änderungen zum
  Commit vorzumerken)

        fonts/glyphicons-halflings-regular.svg
        fonts/glyphicons-halflings-regular.ttf
        fonts/glyphicons-halflings-regular.woff
        js/
```

git add

```
$ git add fonts/ js/
```

Alternativ:

```
$ git add -A
```

git add

```
$ git status
Auf Branch master

Initialer Commit

zum Commit vorgemerkte Änderungen:
  (benutzen Sie "git rm --cached <Datei>..." zum Entfernen
  aus der Staging-Area)

        neue Datei:     css/bootstrap-theme.css
        neue Datei:     css/bootstrap-theme.css.map
        neue Datei:     css/bootstrap-theme.min.css
        neue Datei:     css/bootstrap.css
        neue Datei:     css/bootstrap.css.map
        neue Datei:     css/bootstrap.min.css
        neue Datei:     fonts/glyphicons-halflings-regular.eot
        neue Datei:     fonts/glyphicons-halflings-regular.svg
        neue Datei:     fonts/glyphicons-halflings-regular.ttf
        neue Datei:     fonts/glyphicons-halflings-regular.woff
        neue Datei:     js/bootstrap.js
        neue Datei:     js/bootstrap.min.js
```

git commit

```
$ git commit -m "Bootstrap hinzugefügt."
  [master (Basis-Commit) 7f1c942] Bootstrap hinzugefügt.
  12 files changed, 9006 insertions(+)
  create mode 100644 css/bootstrap-theme.css
  create mode 100644 css/bootstrap-theme.css.map
  create mode 100644 css/bootstrap-theme.min.css
  create mode 100644 css/bootstrap.css
  create mode 100644 css/bootstrap.css.map
  create mode 100644 css/bootstrap.min.css
  create mode 100644 fonts/glyphicons-halflings-regular.eot
  create mode 100644 fonts/glyphicons-halflings-regular.svg
  create mode 100644 fonts/glyphicons-halflings-regular.ttf
  create mode 100644 fonts/glyphicons-halflings-regular.woff
  create mode 100644 js/bootstrap.js
  create mode 100644 js/bootstrap.min.js
```

git status

```
$ git status
# Auf Zweig master
nichts einzutragen, Arbeitsverzeichnis sauber
```

git log

```
$ git log
commit 7f1c942a8275fdeab84ebee61e6fe43a6d48e888
Author: Sujeevan Vijayakumaran <mail@svij.org>
Date:   Sun Jul 20 17:24:13 2014 +0200

    Bootstrap hinzugefügt.
```

Inhalt hinzufügen

index.html hinzufügen:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

git add && git commit

```
$ git add index.html
$ git commit -m "index.html hinzugefügt."
[master 4cc7ce4] index.html hinzugefügt.
1 file changed, 21 insertions(+)
create mode 100644 index.html
```

git log

```
$ git log

commit 4cc7ce45fb1a73d10325b465062d1ffa3435702f
Author: Sujeevan Vijayakumaran <mail@svij.org>
Date:   Sun Jul 20 17:37:51 2014 +0200

    index.html hinzugefügt.

commit 7f1c942a8275fdeab84ebee61e6fe43a6d48e888
Author: Sujeevan Vijayakumaran <mail@svij.org>
Date:   Sun Jul 20 17:24:13 2014 +0200

    Bootstrap hinzugefügt.
```

Inhalte verändern

Title-Tag verändern:
<title>Webseite mit Git</title>
Begrüßung im H1-Tag:
<h1>Hallo Git!</h1>

git status

```
$ git status
Auf Branch master
Änderungen, die nicht zum Commit vorgemerkt sind:
  (benutzen Sie "git add <Datei>..." um die Änderungen
  zum Commit vorzumerken)
  (benutzen Sie "git checkout -- <Datei>..." um die
  Änderungen im Arbeitsverzeichnis zu verwerfen)

        geändert:       index.html

        keine Änderungen zum Commit vorgemerkt (benutzen
        Sie "git add" und/oder "git commit -a")
```

git diff

```
$ git diff

  diff --git a/index.html b/index.html
  index 7a050c7..ea3b0af 100644
  --- a/index.html
  +++ b/index.html
  @@ -5,13 +5,13 @@
      <meta charset="utf-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <meta name="viewport" content="width=device-width, initial-scale=1">
  -    <title>Bootstrap 101 Template</title>
  +    <title>Webseite mit Git</title>

      <!-- Bootstrap -->
      <link href="css/bootstrap.min.css" rel="stylesheet">
    </head>
    <body>
  -    <h1>Hello, world!</h1>
  +    <h1>Hallo Git!</h1>

      <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
      <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
```

git add && git commit

```
$ git add index.html
$ git commit -m "Titel und Überschrift angepasst."
[master 24e65af] Titel und Überschrift angepasst.
1 file changed, 2 insertions(+), 2 deletions(-)
```

Branching-Modell

Abbildung: Git-Logo

Warum Branches?

- Stable
- Develop
- Develop+1
- Hot-Fix
- Feature/XY

Branches anzeigen

```
$ git branch
* master
```

Branch anlegen

```
$ git branch menu

$ git branch
* master
  menu
```

Branch wechseln

```
$ git checkout menu
Gewechselt zu Branch 'menu'
```

Branch anlegen & wechseln

```
$ git checkout -b menu
```

Zweiten Branch anlegen & wechseln

```
$ git checkout -b content
```

Neuer Commit

Füge beliebigen Inhalt unterhalb der <h1>-Überschrift hinzu!

```
<p>
  Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
  sed diam nonumy eirmod tempor invidunt ut labore et
  dolore magna aliquyam erat, sed diam voluptua. At
  vero eos et accusam et justo duo dolores et ea
  rebum. Stet clita kasd gubergren, no sea takimata
  sanctus est Lorem ipsum dolor sit amet.
</p>
```

Ein Menü hinzufügen

```
$ git checkout menu
```

```
<nav class="navbar navbar-default" role="navigation">
  <div class="container-fluid">
    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Link</a></li>
        <li><a href="#">Link</a></li>
      </ul>
    </div>
  </div>
</nav>
```

Änderungen committen

```
$ git add index.html
$ git commit -m "Bootstrap-Beispiel-Menü hinzugefügt"
```

Der erste Merge

```
$ git checkout master
$ git merge menu
  Aktualisiere 24e65af..c3cf413
  Fast-forward
  index.html | 10 ++++++++++
  1 file changed, 10 insertions(+)
```

Der zweite Merge

```
$ git merge content
  Merge branch 'content'

  # Bitte geben Sie eine Commit-Beschreibung ein um zu erklären, warum dieser
  # Merge erforderlich ist, insbesondere wenn es einen aktualisierten
  # Upstream-Branch mit einem Thema-Branch zusammenführt.
  #
  # Zeilen beginnend mit '#' werden ignoriert, und eine leere Beschreibung
  # bricht den Commit ab.

  ...

  automatischer Merge von index.html
  Merge made by the 'recursive' strategy.
    index.html | 3 +++
    1 file changed, 3 insertions(+)
```

Branch „titel“ anlegen

```
$ git checkout -b content
```

Inhalt verändern

<h1>-Überschrift anpassen!

```
$ git add index.html
$ git commit -m "Titel für den Merge-Konflikt"
[titel 420e0ae] Titel für den Merge-Konflikt
1 file changed, 1 insertion(+), 1 deletion(-)
```

Inhalt in 'master' verändern

```
$ git checkout master
<h1>-Überschrift, diesmal anders, anpassen!

$ git add index.html
$ git commit -m "Neuer Titel"
[master 9cb085b] Neuer Titel
1 file changed, 1 insertion(+), 1 deletion(-)
```

Der Merge-Konflikt

```
$ git merge titel
automatischer Merge von index.html
KONFLIKT (Inhalt): Merge-Konflikt in index.html
Automatischer Merge fehlgeschlagen; beheben
Sie die Konflikte und committen Sie dann
das Ergebnis.
```

Der Merge-Konflikt

```
$ git status
Auf Branch master
Sie haben nicht zusammengeführte Pfade.
(beheben Sie die Konflikte und führen Sie "git commit" aus)

Nicht zusammengeführte Pfade:
(benutzen Sie "git add/rm <Datei>..." um die Auflösung zu markieren)

      von beiden geändert:    index.html

keine Änderungen zum Commit vorgemerkt (benutzen Sie "git add" und/oder "git commit -a")
```

Konflikt ansehen

```
In index.html:
<<<<<<< HEAD
    <h1>Hallo!</h1>
=======
    <h1>Hallo Merge-Konflikt!</h1>
>>>>>>> titel
```

Konflikt lösen

```
$ git add index.html
$ git status
Auf Branch master
Alle Konflikte sind behoben, aber Sie sind immer noch
beim Merge.
  (benutzen Sie "git commit" um den Merge abzuschließen)

nichts zu committen, Arbeitsverzeichnis unverändert
```

Mergen

```
$ git commit
Merge branch 'titel'

Conflicts:
        index.html
#
# Es sieht so aus, als committen Sie einen Merge.
# Falls das nicht korrekt ist, löschen Sie bitte die Datei
#       .git/MERGE_HEAD
# und versuchen Sie es erneut.


# Bitte geben Sie eine Commit-Beschreibung für Ihre Änderungen ein. Zeilen,
# die mit '#' beginnen, werden ignoriert, und eine leere Beschreibung
# bricht den Commit ab.
# Auf Branch master
# Alle Konflikte sind behoben, aber Sie sind immer noch beim Merge.
```

Branches löschen

```
$ git branch -d titel
```

Remote Repositories anlegen

```
$ mkdir ~/Git
$ git clone --bare . ~/Git/Webseite-mit-Git.git
Klone in Bare-Repository '/home/sujee/Git/Webseite-mit-Git.git'...
Fertig.
$ git clone --bare . ~/Git/Er.git
Klone in Bare-Repository '/home/sujee/Git/Er.git'...
Fertig.
$ git clone --bare . ~/Git/Ich.git
Klone in Bare-Repository '/home/sujee/Git/Ich.git'...
```

Remote Repository hinzufügen

```
$ git remote add origin ~/Git/ich.git
```

Weitere Remotes hinzufügen

```
$ git remote add upstream ~/Git/Webseite-mit-Git.git
$ git remote add er ~/Git/Er.git
```

Remote-Repository fetchen

```
$ git fetch upstream
Von /home/sujee/Git/Webseite-mit-Git
 * [neuer Branch]     content    -> upstream/content
 * [neuer Branch]     master     -> upstream/master
 * [neuer Branch]     menu       -> upstream/menu
```

Alle Remotes updaten

```
$ git remote update
Fordere an von upstream
Fordere an von er
Von /home/sujee/Git/Er
 * [neuer Branch]     content    -> er/content
 * [neuer Branch]     master     -> er/master
 * [neuer Branch]     menu       -> er/menu
Fordere an von origin
Von /home/sujee/Git/ich
 * [neuer Branch]     content    -> origin/content
 * [neuer Branch]     master     -> origin/master
 * [neuer Branch]     menu       -> origin/menu
```

Branch zum Remote pushen

```
$ git push origin master
Zähle Objekte: 7, Fertig.
Delta compression using up to 4 threads.
Komprimiere Objekte: 100% (7/7), Fertig.
Schreibe Objekte: 100% (7/7), 885 bytes | 0 bytes/s, Fertig.
Total 7 (delta 3), reused 0 (delta 0)
To /home/sujee/Git/ich.git
   9769186..bfb4d20  master -> master
```

Details zum Remote-Repository

```
$ git remote show origin
* Remote-Repository origin
  URL zum Abholen: /home/sujee/Git/ich.git
  URL zum Versenden: /home/sujee/Git/ich.git
  Hauptbranch: master
  Remote-Branches:
    content gefolgt
    master  gefolgt
    menu    gefolgt
  Lokale Referenzen konfiguriert für 'git push':
    content versendet nach content (aktuell)
    master  versendet nach master  (aktuell)
    menu    versendet nach menu    (aktuell)
```

git push

```
$ git push --set-upstream origin master
Branch master konfiguriert zum Folgen von Remote-Branch master von origin.
Everything up-to-date
```

git pull

```
$ git branch --set-upstream-to=origin/master master
  Branch master konfiguriert zum Folgen von Remote-Branch master von origin.

$ git pull
```

Remote Branche auschecken

```
$ git checkout --track origin/fix1337
```

Remote Branches löshen

```
$ git push origin :fix1337
```

Git auf dem Server

Fremd-Hosting:

- GitHub
- Bitbucket

Zum Selbst-Hosten:

- Gitosis
- gitolite
- Gitlab

Was ist GitHub?

- Kostenloses Hosting von öffentlichen Git-Repositories
- Kostenpflichtiges Hosting von privaten Git-Repositories
- einfache Möglichkeit zur kollaborativen Entwicklung
- Nutzer- und Organisationsaccounts
- viele OpenSource-Projekte:
  - Django
  - jQuery
  - Perl
  - PHP
  - u.v.m.

GitHub-Education

Tipp für Studenten
GitHub-Education Pack:

- .me Domain (1 Jahr)
- Zugang zu diversen Cloud-Diensten
- 5 private Repositories!

Was ist Gitlab?

- OpenSource GitHub-Klon
- Community-Edition zum Selbsthosten

Fragen

Noch Fragen?

Vielen Dank für die Aufmerksamkeit!
Folien finden sich auf GitHub:
https://github.com/svijee/git-workshop