JAVACREAM

Training
Consulting
Projectmanagement

GIT

Vorstellungsrunde

JAVACREAM
Training
Consulting
Projectmanagement

- Name und Rolle im Unternehmen
- Themenbezogene Vorkenntnisse
- Aktuelle Problemstellung
- Individuelle Zielsetzung

Einführung

Erwartungshaltung an Versionsverwaltungssystem

- Verwaltung von “Meta-Informationen” wird übernommen
  - “wer hat wann was warum gemacht”
- Bestimmte Dateien werden in bestimmten Ständen zu einem Gesamt-Stand gruppiert
- Stände können parallel existieren
- Konsolidierung von Ständen
- Tooling, Historischer Verlauf, Unterschiede in Dateien/Ständen, …
- Gemeinsamer Zugriff durch einen Server (Authentifizierung, …)
- Team-Zusammenarbeit

Realisierung mit GIT

- Verwaltung von “Meta-Informationen” wird übernommen
  - “wer hat wann was warum gemacht”
- Bestimmte Dateien werden in bestimmten Ständen zu einem Gesamt-Stand gruppiert
- Stände können parallel existieren
- Konsolidierung von Ständen
- Tooling, Historischer Verlauf, Unterschiede in Dateien/Ständen, …
  - Konsole
- Gemeinsamer Zugriff durch einen Server (Authentifizierung, …)
- Team-Zusammenarbeit

Realisierung mit GIT, Git-Werkzeuge

- Verwaltung von “Meta-Informationen” wird übernommen
  - “wer hat wann was warum gemacht”
- Bestimmte Dateien werden in bestimmten Ständen zu einem Gesamt-Stand gruppiert
- Stände können parallel existieren
- Konsolidierung von Ständen
- Tooling, Historischer Verlauf, Unterschiede in Dateien/Ständen, …
  - Konsole, Integration ins Betriebssystem, Integration in Editoren und IDEs
- Gemeinsamer Zugriff durch einen Server (Authentifizierung, …)
- Team-Zusammenarbeit

Zu den Git-Werkzeugen

- Desktop
  - TortoiseGit
  - SourceTree
  - …
- PlugIns für Editoren
  - Eclipse
  - Visual Studio
  - Visual Studio Code
  - ...

Realisierung mit GIT, Git-Werkzeuge, Git Server

- Verwaltung von “Meta-Informationen” wird übernommen
  - “wer hat wann was warum gemacht”
- Bestimmte Dateien werden in bestimmten Ständen zu einem Gesamt-Stand gruppiert
- Stände können parallel existieren
- Konsolidierung von Ständen
- Tooling, Historischer Verlauf, Unterschiede in Dateien/Ständen, …
  - Konsole, Integration ins Betriebssystem, Integration in Editoren und IDEs, Web Console
- Gemeinsamer Zugriff durch einen Server (Authentifizierung, …)
- Team-Zusammenarbeit

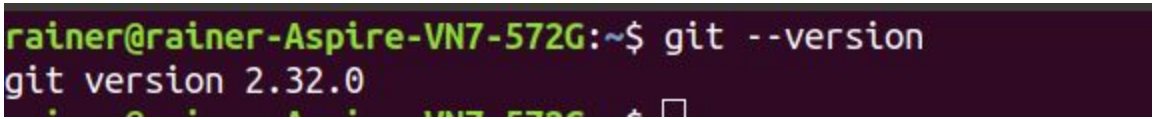
Git Server

- GitHub
  - Server laufen in der Microsoft-Cloud
  - Öffentliche Ablage ist kostenlos, private Bereiche Lizenz-pflichtig
- BitBucket
  - Atlassian
  - Cloud-Service + Betrieb auf eigenen Servern
- GitLab
  - gitlab.com
  - Cloud-Service + Betrieb auf eigenen Servern
- Azure DevOps
  - Microsoft-Cloud

First Contact

Git Bash

- Terminal-Fenster mit Git-Unterstützung
- Das Kommando “git” steht hierin zur Verfügung
  - Das ist **kein simples Command Line Interface**, das mit einem Git-Server kommuniziert
    - Es gibt keinen laufenden Git-Server-Prozess, kein Dämon, …
  - git --version
  -



Der erste Git-Befehl: config

- git config <scope> <key-hierarchie> <value>
  - <scope>
    - local
    - global (User-Profile)
    - system
  - <key-hierarchie>
    - “.” trennt die Hierarchie-Ebenen
  - <value>
    - irgendwas, Leerzeichen etc. aber in Anführungszeichen setzen
- git config --global user.name “Rainer Sawitzki”
- git config --global user.email training@rainer-sawitzki.de
- Auslesen git config --get user.name

Erstes Arbeiten mit Git

Einrichten eines Git-Projektverzeichnisses


```
mkdir first
cd first
```

```
git init
```

```
git status
        Fehlerfrei
```

Normales Verzeichnis -> Git-Projektverzeichnis

Git Workspace

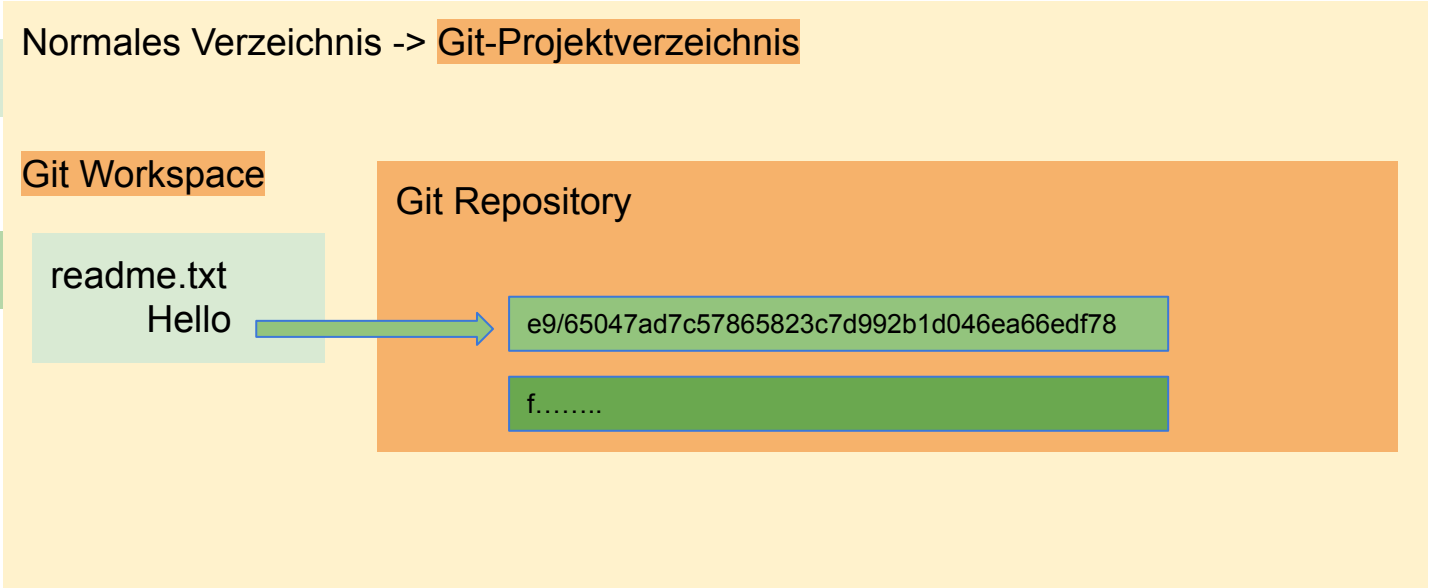
Git Repository

Hinweis:
In der Praxis entspricht diese Sequenz einem
git clone server-repo first

Hinzufügen von Informationen


```
echo Hello > readme.txt
```

```
git add readme.txt
```



Exkurs

- Was ist diese Datei “e9/65047ad7c57865823c7d992b1d046ea66edf78”?
  - Binärformat, das heißt Informationen sind in dieser Datei abgelegt worden
  - Der Dateiname ist
    - ~~UID, weltweit eindeutiger Zufallswert~~ oder
    - Ein berechneter Hash-Wert, berechnet aus dem Inhalt “Hello”

Interpretation

- e9/65047ad7c57865823c7d992b1d046ea66edf78
- Wenn in irgendeinem Git-Repository dieser Hash-Wert existiert ist es mit höchster Wahrscheinlichkeit eine Datei mit dem Inhalt “Hello”
  - Wahrscheinlichkeit einer Kollision ist absurd gering

Commits definieren Stände


```
echo Hello > readme.txt
```

```
git add readme.txt
```

```
git commit -m “warum?”
```

Normales Verzeichnis -> Git-Projektverzeichnis

Git Workspace

readme.txt
Hello

Git Repository

Index / Staging Area

e9/65047ad7c57865823c7d992b1d046ea66edf78

f……..

Commit

Objekt

Aufbau des Commit-Objekts

- Liste der Dateien / Informationen, die zu diesem Stand=Commit gehören
- Committer
  - user.name + user.email
- Timestamp
- Commit-Message
  - Angabe über Option git commit -m “...”
  - Ohne Option öffnet sich ein Editor-Fenster
    - vim (Drücke “i” für den Insert-Modus, Fertig: ESC, Eingabe von :wq
    - nano
    - git config --global core.editor notepad

Interpretation

- Ist in einem Repository ein Commit mit dem Hash ee23e95a40724fb1ad5b119d2ed9e8f7c069813e vorhanden:
  - Rainer Sawitzki hat am um eine Commit erstellt mit der Message add content und den Dateien readme.txt(Hello) content.txt(Hugo)