



HOCHSCHULE FÜR
TECHNIK UND WIRTSCHAFT
DRESDEN
UNIVERSITY OF APPLIED SCIENCES

Versionsverwaltung mit git

Simon Retsch

Jonatan Strube

13.12.2017

1. Prinzip Versionsverwaltung

2. Grundlagen

3. Anbieter

4. Apps

5. Vorführung

Eigenschaften:

- Sicherung
- Übersicht
- Organisation mehrerer Bearbeiter

- add
- commit
- push
- pull
- branch
- merge



Master Projekt (Master branch)

Lukas erstellt branch von Master
commit von Lukas

Johanna erstellt branch von Lukas

Master merged seinen branch mit Lukas branch

Master merged seinen branch mit Johannas branch

Versionierung sinnvoll bei:

- Textdateien

Nicht sinnvoll bei:

- Binärdateien

- Schwarze Liste von Dateien/Ordernern

- Binäres `*.exe`
- Von Programmen `.vscode/`
- Geheimes `my_password.txt`

- Weiße Liste

- Ausnahmen zur schwarzen Liste mit `!final/*`

Datei .gitignore:

```
# alle Binärdateien:  
*.exe  
*.out  
# Ordner von VSC:  
.vscode/  
# Geheimes:  
mein_passwort.txt  
bildchen/*  
# Behalte Einzeldatei:  
!bildchen/goodprof.png
```



```
# alle Binärdateien:  
*.exe  
*.out  
# Ordner von VSC:  
.vscode/  
# Geheimes:  
mein_passwort.txt  
bildchen/*  
# Behalte Einzeldatei:  
!bildchen/goodprof.png
```

```
repository/  
├── .git/  
├── bildchen/  
│   ├── badprof.png  
│   └── goodprof.png  
├── programm/  
│   ├── .vscode/  
│   │   └── ...  
│   ├── readme.md  
│   ├── virus.c  
│   ├── virus_unix.out  
│   └── virus_win.exe  
├── .gitignore  
├── deren_passwoerter.txt  
├── mein_passwort.txt  
├── no_virus.exe  
└── virus.png
```

Commit Nachricht besteht aus:

- Thema / Titel (*subject*)
Kurze Beschreibung.
- Rumpf / Beschreibung (*body*)
Was wurde warum geändert (nicht *wie*)

Richtlinien Titel

- Kurz (50, max. 72 Zeichen)
- Mit Großbuchstabe beginnen
- Kein Punkt am Ende
- Imperativ

Negativbeispiel

Erweiterung Keylogger, der alles Aufgezeichnete in eine Datei im selben Ordner speichert.

Es wurde ein Keylogger durch memory injection implementiert. In Zeilen 50-100 der Codedatei kann man gut sehen, wie die durch einen buffer overflow jedes Betriebssystem komprimiert werden kann.

Mehr Passwörter! Yay.

Beispielnachricht

Erweitere Virus um Keylogger

Es wurde ein Keylogger implementiert, der alle Nutzereingaben aufzeichnet und speichert.

Dadurch können mehr Passwörter ausgelesen werden.

- GitHub
- GitLab
- Bitbucket

Kostenfreie Version:

Kategorien	Bitbucket	GitHub	GitLab
Public	ja	ja	ja
Private Repositories	ja	nein	ja
Teams	ja (limit: 5)	ja	ja

- SmartGit, SourceTree, ...
- GitHub Desktop
- Integriert in Visual Studio Code

- Umfassende GUIs
- Erweiterte Funktionalitäten über GUI:
 - cherry picking
 - Konfliktlösungen
 - ...

repository - C:\Users\jstrube\Documents\SmartGit\repository - SmartGit 17.1.3 (for non-commercial use only)

Repository Edit View Remote Local Branch Query Changes Tools Window Help

Pull Sync Push Git-Flow Merge Commit Stage Index Editor Unstage Remove Abort Discard Delete Log Blame Investigate

Repositories x Files x 12 files hidden File Filter

Name	State	Relative Directory
deren_passwoerter.txt	Modified	

Changes x

Journal x

Message	Author	Date
origin<master>		

Ready 1 Directory

The screenshot shows the SourceTree application window with the following elements:

- Menu Bar:** Datei, Bearbeiten, Ansicht, Projektarchiv, Aktionen, Tools, Hilfe.
- Toolbar:** Commit, Push, Pull, Anfordern, Zweig, Zusammenführen, Stash, Verwerfen, Tag, Git Flow, Eingabeaufforderung, Explorer, Einst.
- Left Panel (Repository Structure):**
 - DATEI STATUS
 - Arbeitskopie
 - ZWEIGE
 - experiment
 - master**
 - TAGS
 - REMOTES
 - STASHS
- Main Area (File Status):**
 - Header: Ausstehende Dateien, sortiert nach Dateistatus. Suchen [] []
 - Buttons: Alles nicht mehr vormerken, Ausgewählte nicht mehr vormerken
 - Section: Nicht vorgemerkte Dateien
 - Buttons: Alles vormerken, Ausgewählte vormerken
 - File: deren_passwoerter.txt
- Commit Section:**
 - Commit-Einstellungen []
 - Text input field
 - Änderungen sofort nach origin/master pushen
 - Commit button
- Bottom Bar:** Dateistatus, Log / Verlauf, Suchen

- Sehr eingeschränkte Funktionalität
- Für Alltag ausreichend

The screenshot shows the GitHub Desktop application window. The top bar includes a menu (File, Edit, View, Repository, Branch, Help) and status information: 'Current repository: repository', 'Current branch: master', and 'Fetch origin: Last fetched just now'. The main area is split into three panes:

- Changes:** Shows '1 changed file' and 'deren_passwoerter.txt' with a yellow square icon.
- History:** Displays a commit log for 'deren_passwoerter.txt'. The log shows a sequence of commits from line 7 to 18. Commits 10, 16, and 18 are highlighted in blue, indicating they are part of the current commit. Commit 10 is a merge commit with the message '-<<<<<<< HEAD'. Commit 16 has the message '-*****'. Commit 18 has the message '->>>>>> experiment'. Other commits include 'user7: anderes', 'user8: 1_2_3', 'user9: afjdslk', 'user10: asdf', 'user11: anderes', 'user12: neues', 'user13: klein', and 'user14: riesig'. The commit for line 17 is 'user9000: ja'.
- Summary:** A form for the commit message with a 'Summary' field and a larger 'Description' field. A blue 'Commit to master' button is at the bottom.

Features:

- Einbindung in Entwicklungsumgebung
 - Direkter Import (clone) von Git Repositories
 - Grundlegende Befehle sind verfügbar
- Anzeige aktuell geänderter Passagen im Editor

Vorführung zeigt git-Nutzung in Visual Studio Code:

- Veranschaulichung:
 - *Pull*
 - *Commit*
 - *Push*
- Konflikte Lösen bei parallelem Bearbeiten (ohne *pull* vor *commit*)
- In anderen *branch* wechseln/arbeiten und zurück wechseln (*Checkout*)
- *Branch* zum *master* mergen

Grundlagen

- <https://gist.github.com/robertpainsi/b632364184e70900af4ab688decf6f53>
- <https://wiki.openstack.org/wiki/GitCommitMessages>
- <https://chris.beams.io/posts/git-commit/>

Anbieter

- <https://medium.com/flow-ci/github-vs-bitbucket-vs-gitlab-vs-coding-7cf2b43888a1>

Apps

- <http://www.syntevo.com/smartgit/>
- <https://www.sourcetreeapp.com/>
- <https://desktop.github.com/>