

Ankündigungen

Zürich University
of Applied Sciences



~~Brown Bag Seminar~~
Datalab Seminar



Sonst ändert sich nix.

Und wie immer auf der Suche nach Freiwilligen:

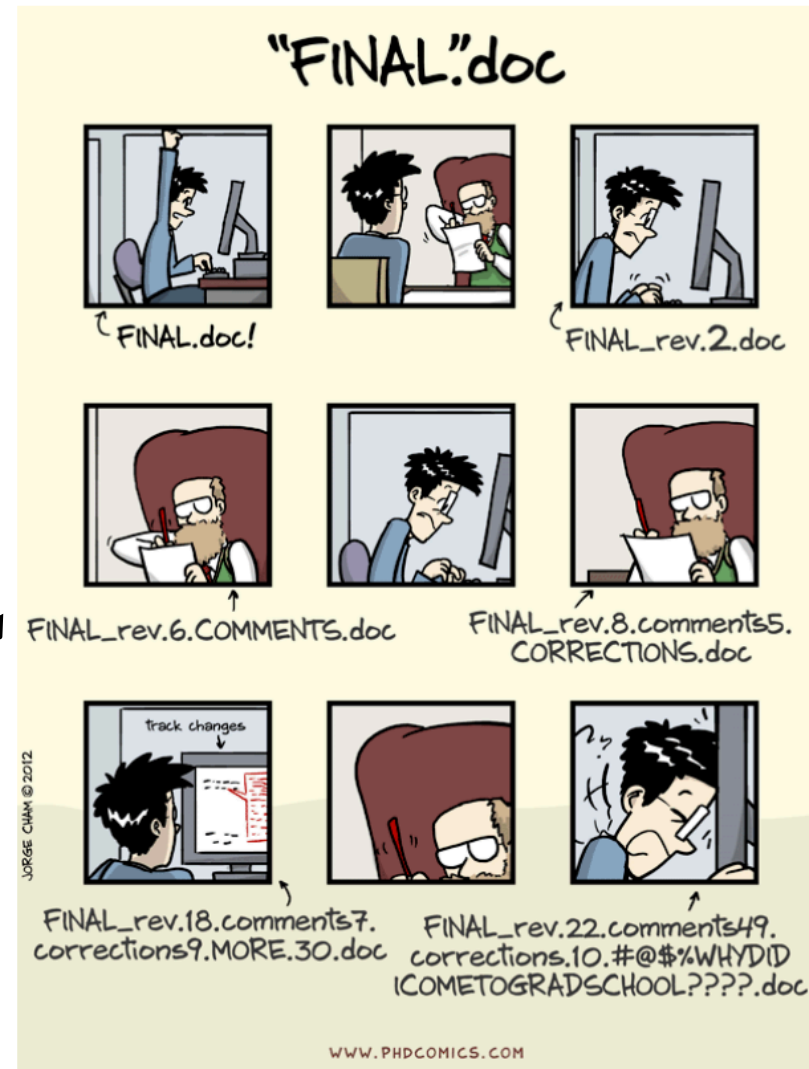
Datalab Seminars

Version Control with git/github

Oliver Dürr

17 September 2014

Thinks get worse if you
work in a team



<http://www.phdcomics.com/comics/archive.php?comid=1531>

Disclaimer

Zürich University
of Applied Sciences



A good datalab tradition:

...I am not an Expert!

And sorry for the mix of English / German. I first planned to do the talk in German.

- Introduction git / github
- Basic Concept
- Demo
- A bit more advanced stuff
- Working in a team
 - Shared repository
 - Fork & pull cooperation
- Diskussion

Was ist git / github

Zürich University
of Applied Sciences

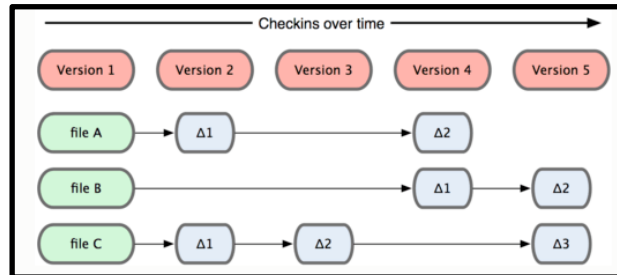


- git ist Tool zur Versionskontrolle (wie SVN, ...)
 - Entstanden aus Linux Kernel Gemeinde (Linus Thorvald 2005). SVN reichte nicht und es gab Streit mit kommerziellen Anbieter.
 - Wichtig git arbeitet mit einem **lokalen Repository** und ist so selbstständig und Hölle schnell
- Github ist host für gitrepositories
 - Frei für public projects
 - Nette Oberfläche zum Browsen
- ZHAW [hat auch git-repository](#)
 - Scheint eine Version von github zu sein (alles was für github gilt auch für ZHAW)
 - Etwas weniger performant? Keine Kooperation mit externen
- Bitbucket
 - Frei auch für private projekte



- Schnelle Einführung
 - <http://rogerdudler.github.io/git-guide/index.de.html>
- Hilfe von github
 - <https://help.github.com/categories/54/articles>
- Kochrezeptartig
 - <http://www-cs-students.stanford.edu/~blynn/gitmagic/>
- Buch (Pro Git, etwas Staubig)
 - <http://git-scm.com/book/en/>
- Commands (Sammlung)
 - <http://www.git-tower.com/blog/git-cheat-sheet-detail/>

Basic Concept, difference to SVM

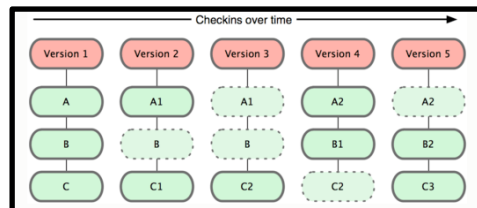


SVN

History only on Server

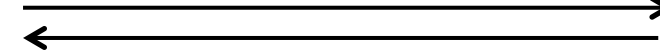
Clients *commit* and *update*

Git



Cont 1

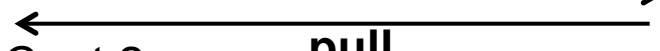
push



Server Rep. nothing special

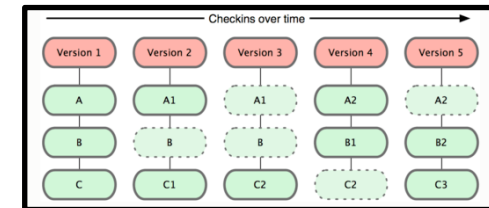
pull

push



Cont 2

pull



Should contain always the latest copy

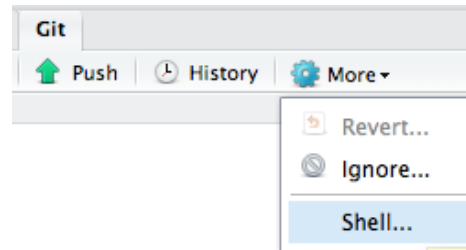
All machines have a full local copy: lightening fast (also due to architecture).

Committing commits to the local copy

RStudio and command-line

- No problem to mix command-line with GUI
 - Prob. also true for other IDE as pycharm, eclipse, intelliJ

- Command-line with



- Command-lines are good to explain the functionality unambiguously
 - `> git pull #Unambiguous`
- Git much better supported then SVM

Create a git-repositories
(local) and link it to github.
in < 5 minutes

- Create local repository
 - Command-line or IDE RStudio
 - Do a little work on the local repository
 - And check in
 - Create an new and empty empty repository on GitHub
- Link and upload existing local directory

Prerequisites

- Command-line Tool for git
 - Windows: <http://msysgit.github.io/>
 - OSX: <http://code.google.com/p/git-osx-installer/>
 - Debian/Ubuntu: `sudo apt-get install git-core`
 - Fedora/RedHat: `sudo yum install git-core`
- A Recent version of Rstudio
 - See <https://support.rstudio.com/...> how to integrate git into RStudio

Erstellen eines (lokalen) Repositories

Zürich University
of Applied Sciences




```
touch README.md
```

```
git init #Creates a repository (Rstudio checkbox)
```

```
git add README.md #Adds README.md to repository (local)
```

```
git commit -m "first commit"
```

Erstellen eines lokalen Repositories mit RStudio



New Project

Back Create R Package

Type: Package Package name: idpedu

Create package based on source files:


Add... Remove

Create project as subdirectory of: ~/Documents/rpackages Browse...

☒ Create a git repository

☐ Open in new window Create Project Cancel

Für R-Projekte und Packages,
einfach checken.



New Project

Back Create New Project

Directory name:

Create project as subdirectory of: ~/Dropbox/_ZHAW/WaST3/WaST3.HS14 Browse...

☐ Create a git repository

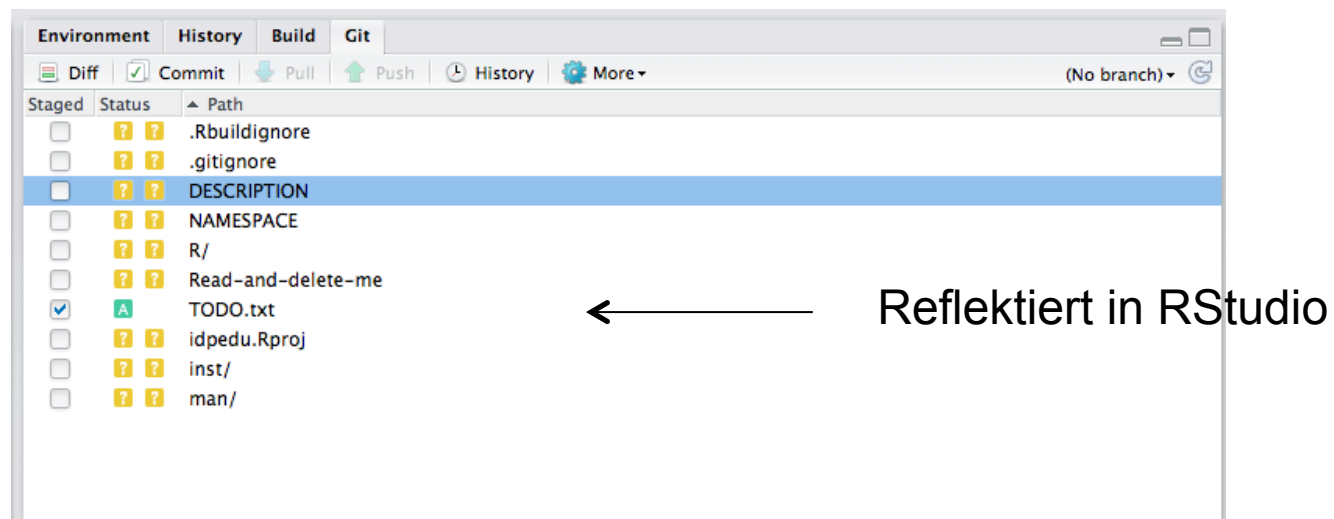
☐ Open in new window Create Project Cancel

Hinzufügen von Dateien (lokal) / RStudio

Zürich University
of Applied Sciences



`git add TODO.txt` ← Lokales Hinzufügen von der Konsole



`git commit -m "first commit"` ← **Lokales Committen**

Github (Erstellen eines Repositories)

Zurich University
of Applied Sciences



Owner: Oliver4242 / Repository name: idpedu ✓

Great repository names are idpedu. How about yolo-octo-happiness.

Description (optional): Usefull tools for teaching statistics

☒ Public
Anyone can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

☐ Initialize this repository with a README
This will allow you to git clone the repository immediately. Skip this step if you have already run git init locally.

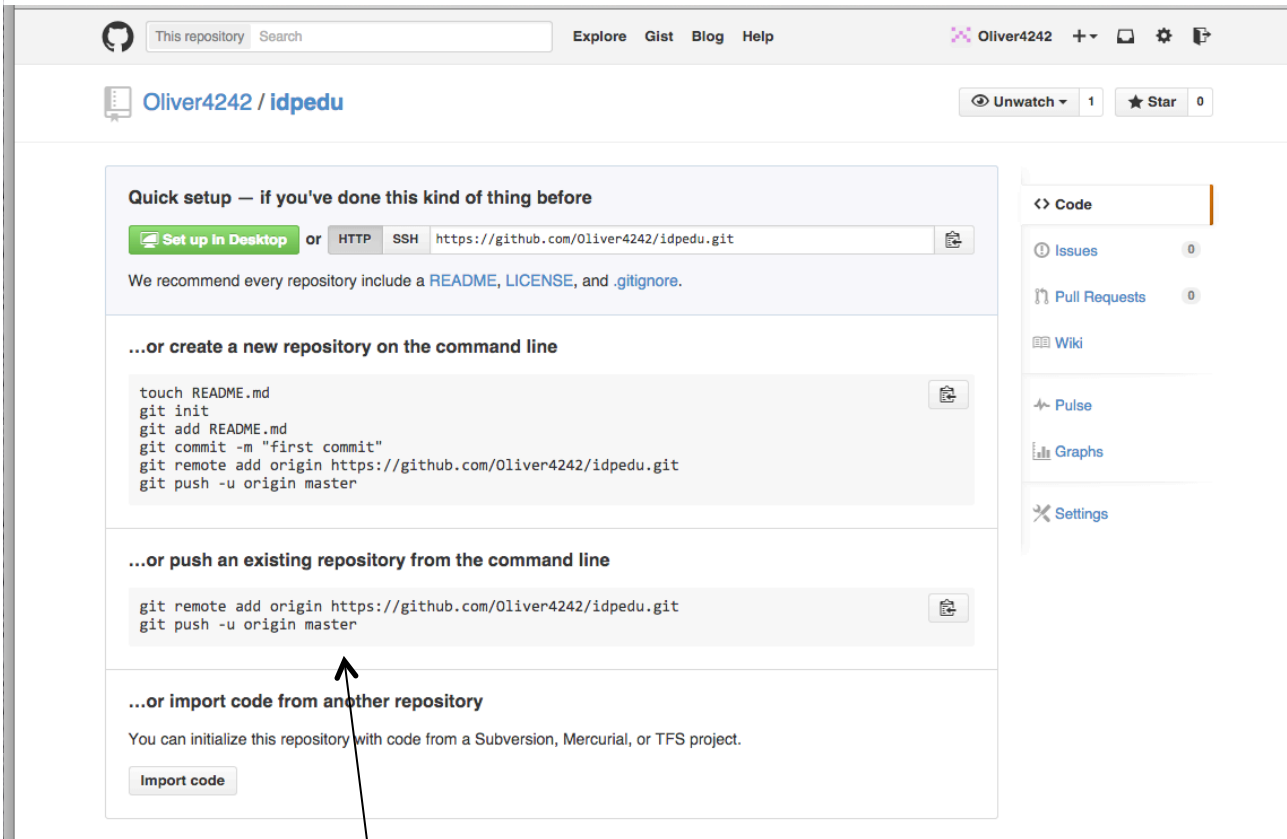
Add .gitignore: None | Add a license: None ⓘ

Create repository

Wenn man schon eines hat dann leer lassen. (Wie hier)

Github (Erstellen eines Repositories)

Zürich University
of Applied Sciences



Neues (lokales)
Repository anlegen und
hochladen (push)

Bestehendes (locales)
hochladen

Einfach copy & paste ...

“Verbinden” des lokalen Repositories mit github und hochladen (push)

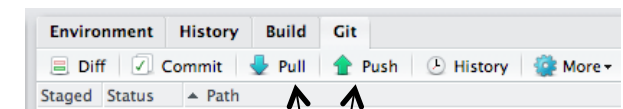
Zürich University
of Applied Sciences



```
git remote add origin https://github.com/Oliver4242/idpedu.git  
git push -u origin master
```

Das Repository liegt nun auf github.

Rstudio hat's auch gemerkt



Nicht mehr grau

A side remark for the R users

Zürich University
of Applied Sciences



An R-package uploaded to github can be installed on other machines via

```
library(devtools) # You might need to install devtools first  
install_github("Oliver4242/idpedu")  
library(idpedu)
```

No compiling or other issues. Just commit & push to github.

Lokal (create a git repository)

- Create package with git support
- Stage and Commit Files (initial commit)
- Add a new function (plot)
- Stage and Commit

On Github

- Create new repository (no readme!)

Lokal

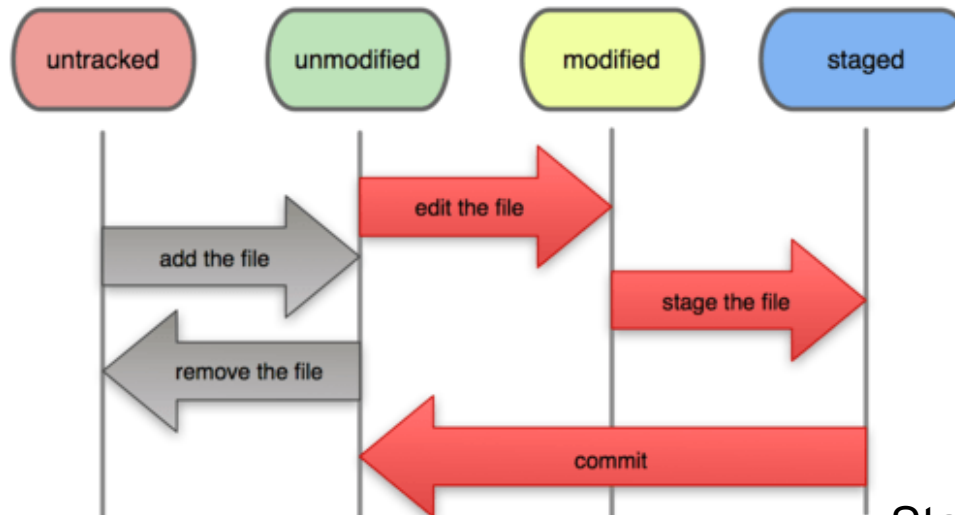
- Link local copy to remote `git remote add origin https://github.com/yourusername/yourrepository.git`
- Upload push local changes `git push -u origin master`
- RStudio has now keys for pull / push

On Github (you find new changes)

Further details

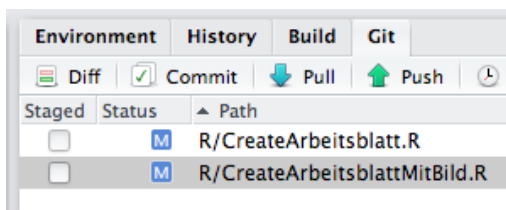
Staging vs Committing

File Status Lifecycle



e.g. in
.gitignore

Staged:
ready to commit



Note: sometimes it's better not to commit
all modified files in one go (changesets).
Reason for staging state

Stageing: `git add R/CreateArbeitsblattMitBild.R`
In Rstudio simple click.

`git add staged` or `adds`
depending on context

For the former SVN users (as me)

Zurich University
of Applied Sciences



Don't forget that committing is just local.

You have to push to upload (see working in a team)

Reverting

Zürich University
of Applied Sciences



```
olivers-mbp:idpedu oli$ git log
```

```
commit cc299be6bf3cb7f864dabbc850a71b3772b8e969
```

```
Author: Oliver4242 <oliver.duerr@gmail.com>
```

```
Date: Sun Sep 14 14:01:58 2014 +0200
```

```
Fixed Bug introduced in the renaming of wast -> idpedu
```

```
commit bbe17dd646ca9dd34c738bee0e83774a5fb2832c
```

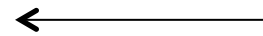
```
Author: Oliver4242 <oliver.duerr@gmail.com>
```

```
Date: Sun Sep 14 13:28:57 2014 +0200
```

```
Some more description
```

```
$ git reset --hard cc299
```

```
$ git revert cc299
```



Eindeutige Hash-Nummer

Auch in RStudio gibt es
eine History.

Ersten paar Ziffern reichen aus.
Neuere Änderungen werden gelöscht.

Changeset cc299 wird gelöscht

Working in a team

Cloning of a repository

New developer enters the game and clones a repository.

```
olivers-mbp:testGit oli$ git clone https://github.com/Oliver4242/idpedu.git
Cloning into 'idpedu'...
remote: Counting objects: 41, done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 41 (delta 7), reused 37 (delta 5)
Unpacking objects: 100% (41/41), done.
Checking connectivity... done.
```

Note to upload changes (push) the new developer needs writing permission

Collaborators

Full access to the repository

This repository currently has no collaborators.

asdasd

Add collaborator

Arbeiten im Team (Workflow)

- Dateien ändern und dann **lokaler commit** (-a = stage und und commit)
`git commit -a -m 'Did some work'`
- Vor dem hochladen checken, was ob es was neues gibt.
pull (= fetch + merge) lädt das aktuelle Repository runter und merged changes (falls es geht).

```
olivers-mbp:idpedu oli$ git pull
```

- Eventuell: Konflikte durch manuelles editieren auflösen und wieder **lokaler commit**
- Hochladen / pushen
`git push`

Workflow (anders als subversion)

commit → pull → push

Ein Konflikt

pull versucht schon Konflikte zu mergen. Allerdings kann es Probleme geben, wenn jemand anders schon dran gearbeitet hat und das file hochgeladen hat.

Konflikt

```
the number of planets are
<<<<<< HEAD
nine
=====
eight
>>>>>> branch-a
```

Konflikt gefixed (manuelles editieren)

```
the number of planets are
nine, or eight, depending on who you ask.
```

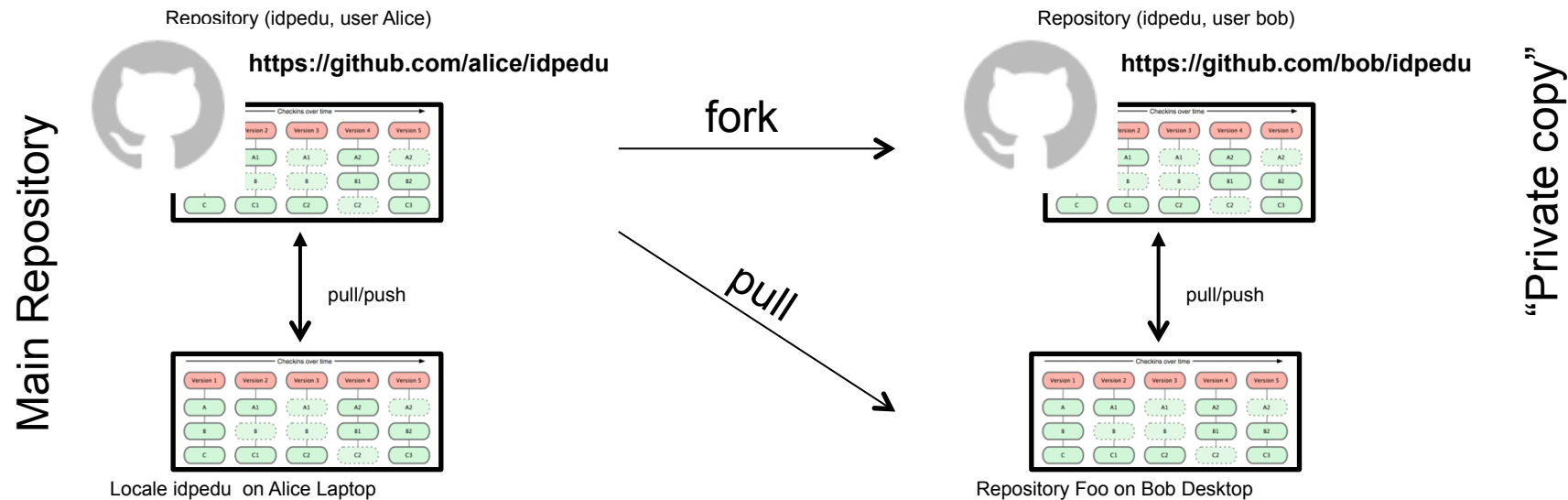
**Konflikt fixen und add, commit (lokal)
(pull, push) zum ins Repository laden.**

Falls Konflikt in einem ganzen File entsteht (removed file Konflikt)

<https://help.github.com/articles/resolving-a-merge-conflict-from-the-command-line>

Fork & pull (github spezifisch)

- Alternative Cooperation. Developers work on their own copy.



Create a local copy from your (bob) repo.

```
git clone git@github.com:bob/idpedu.git
```

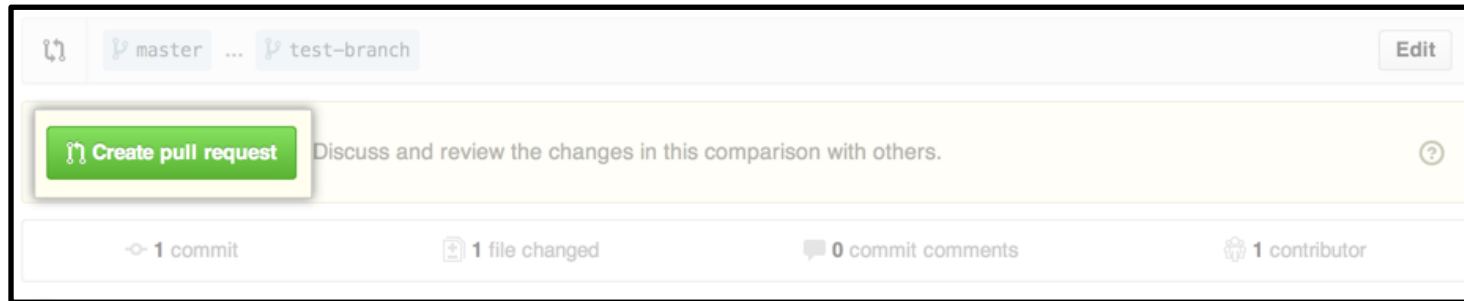
Pull changes from main (alice) repository.

```
git remote add idpedu git://github.com/alice/idpedu.git
git checkout master
git pull alice master
```

How does the Alice benefit from Bobs changes? He is not allowed to push but to do a merge request (pull request) of this repository (on github) and Alice incorporates it.

A pull request

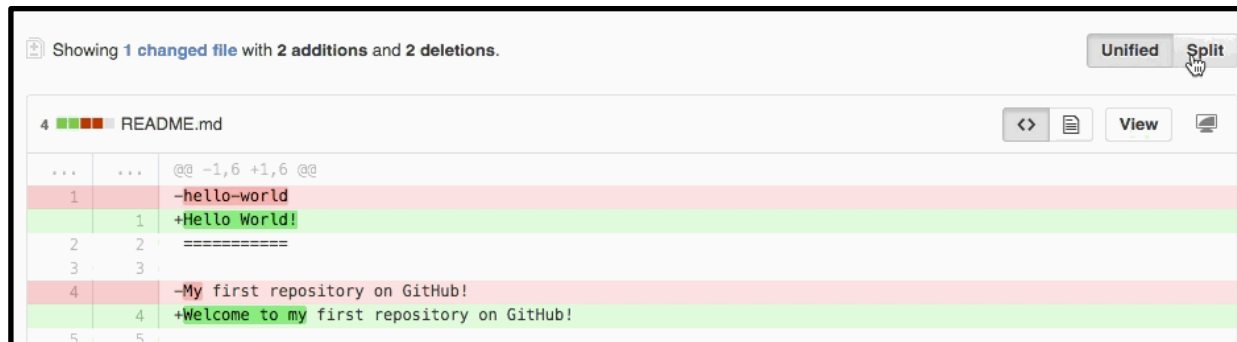
Bob



Alice



New pull request appears

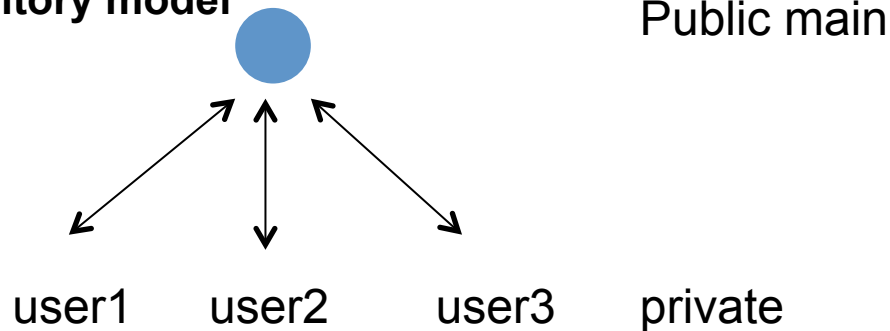


And further webtools
to analyze / merge
the request

See e.g. help.github.com/articles/using-pull-requests

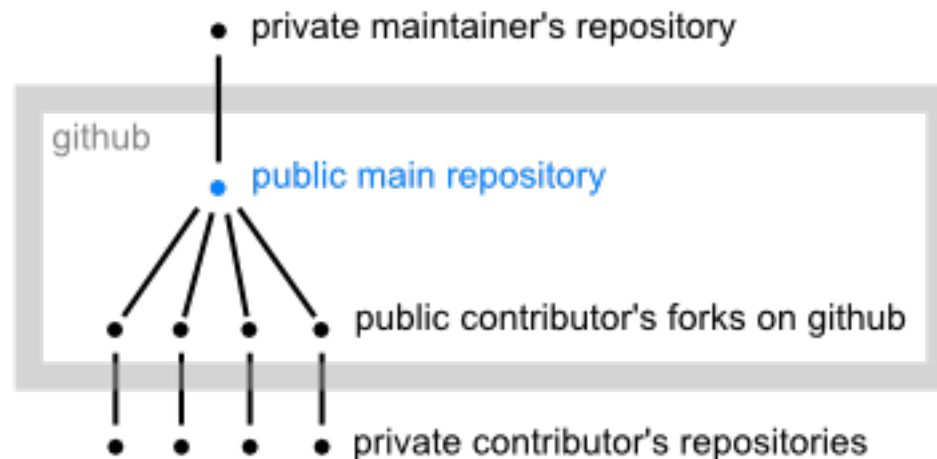
Different collaboration modes

Shared repository model



Alle (contributors) arbeiten auf dem gleichen remote repository.

Fork & Pull model



Each developer has his own public / private repo.

Changes back via *pull-requests*.

Good for large projects

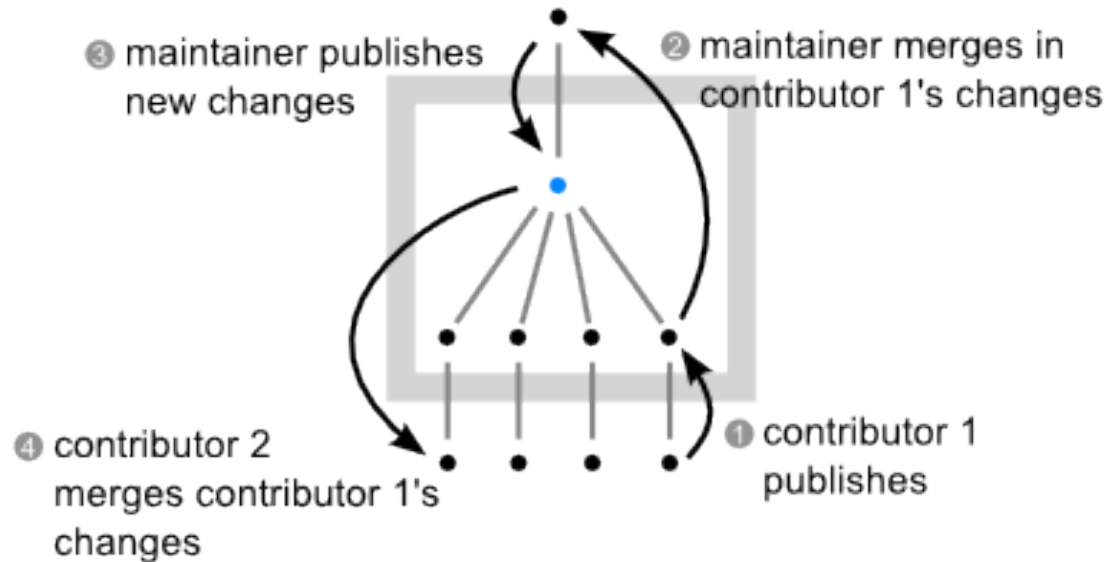
[knitr](#) has 311

[shiny](#) has 301

[ggplot2](#) 265 forks

Workflow for changes

Workflow



Also a great way to bugfix. From github

A great example of using forks to propose changes is for bug fixes. Rather than logging an issue for a bug you've found, you can:

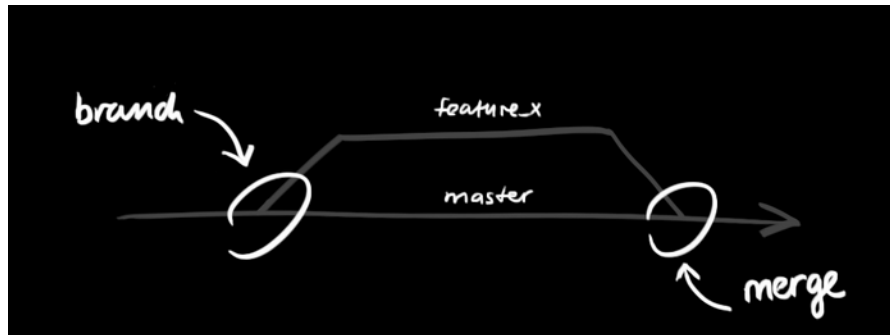
- Fork the repository.
- Make the fix.
- Submit a pull request to the project owner.

Weitere Themen

Zürich University
of Applied Sciences



Branching



...

Webhooks

- Action triggered after checking in
- Continuous Integration with travis
- ...

Maybe room for a talk of a real expert.

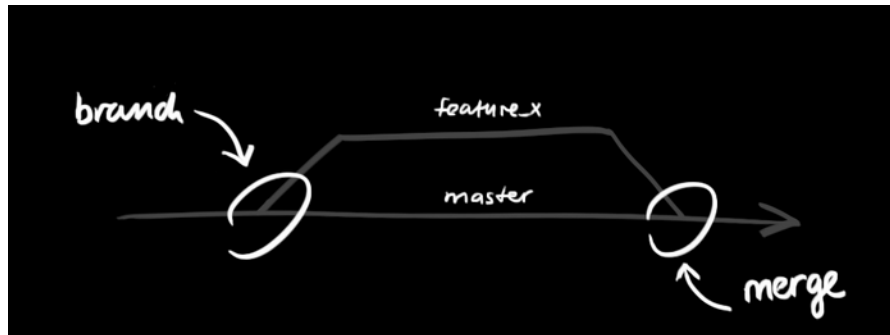
Thanks

Weitere Themen

Zürich University
of Applied Sciences



Branching



...

Webhooks

- Action triggered after checking in
- Continuous Integration with travis
- ...

Maybe room for a talk of a real expert.

Thanks