Was ist git & welches Problem löse ich damit?
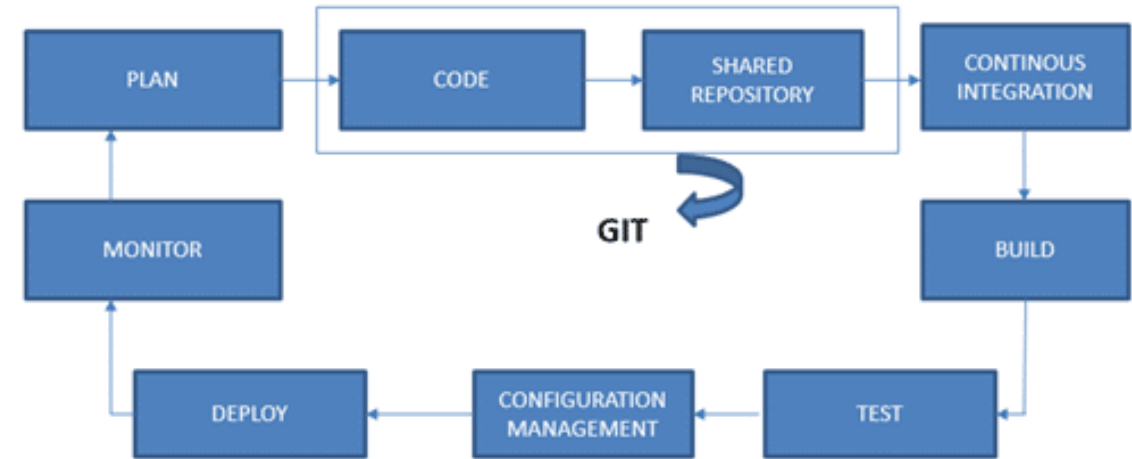

- Wir entwickeln Software (oder irgendetwas anderes)
- Wir entwickeln Software gemeinsam
- Wir machen Fehler (und wollen diese finden und beheben)
- Wir vergessen, was wir getan haben (Dokumentation)
- Wir mögen FOSS

→Wir brauchen **Version Control**

https://intellipaat.com/mediaFiles/2019/03/Process-flow.png

- Kostenlos!
- Plattformübergreifend!
- Weit verbreitet!
- Quasi Infrastruktur!
- Zugang zu moderner Software!
- CLI oder GUI möglich!
- (Quasi) einsteigerfreundlich!
- Gute Doku!

https://xkcd.com/1597/

- FOSS-Community arbeitet mit git

git Alternativen

- Subversion (SVN)
- Mercurial
- …?

Konzepte

- Workflow für die Entwicklung
  - Änderung
  - Überprüfung
  - Speicherung
  - (Veröffentlichung)
  - (Integration + Konfliktlösung)

https://intellipaat.com/mediaFiles/2019/03/Life-Cycle.png

Konzepte

- Workflow für die Entwicklung
- Three-tree architecture
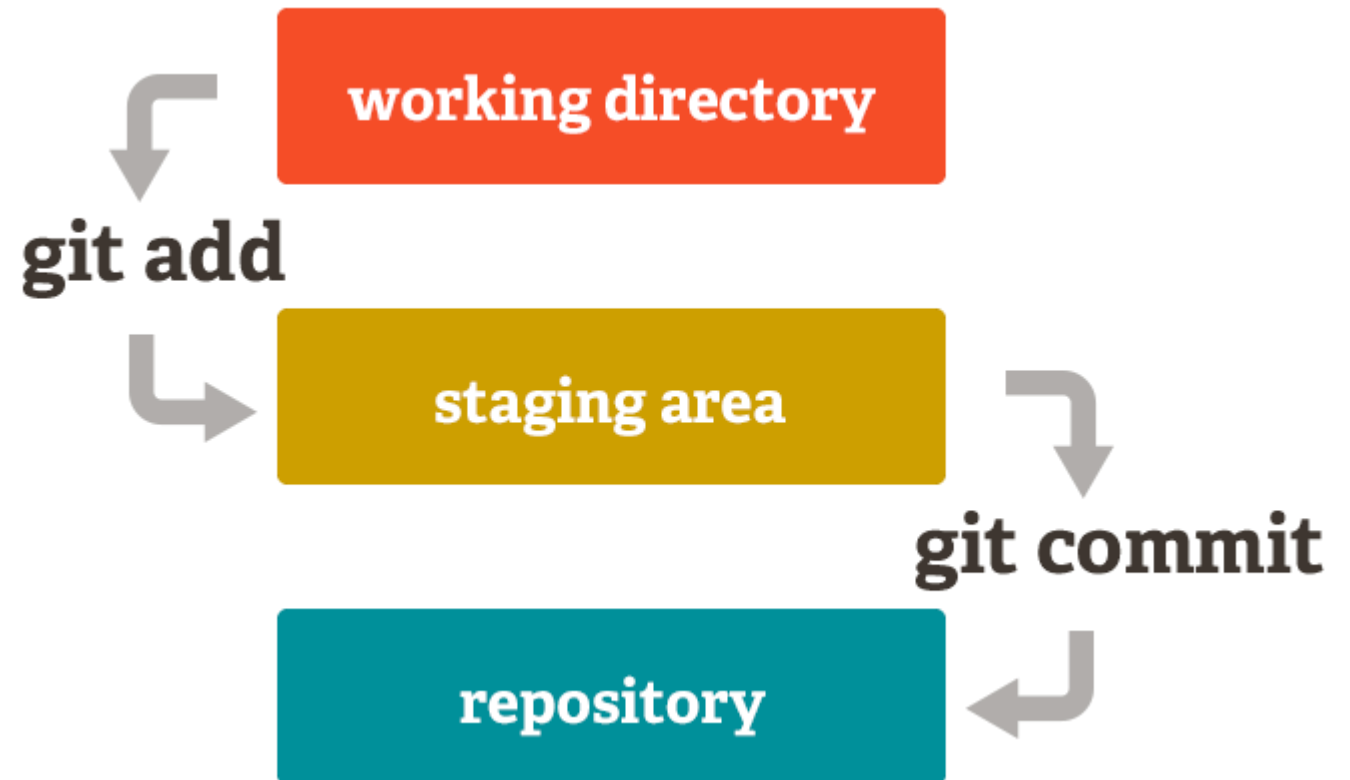
Konzepte

- Workflow für die Entwicklung
- Three-tree architecture
- Branching and merging

Konzepte

- Workflow für die Entwicklung
- Three-tree architecture
- Branching and merging
- Text-basiert (git erkennt Änderungen an Textdateien)
- Alles passiert im Ordner .git/

Das git-Repository

- Dateien!
- Wir müssen uns darum nicht kümmern
- Interaktion über CLI oder GUI
- Dateien über .gitignore ignorieren
- „rm –rf .git/" löst hartnäckige Probleme

```
carl@t14s$ tree -a
.
└── .git
    ├── branches
    ├── config
    ├── description
    ├── HEAD
    ├── hooks
    │   ├── applypatch-msg.sample
    │   ├── commit-msg.sample
    │   ├── fsmonitor-watchman.sample
    │   ├── post-update.sample
    │   ├── pre-applypatch.sample
    │   ├── pre-commit.sample
    │   ├── pre-merge-commit.sample
    │   ├── prepare-commit-msg.sample
    │   ├── pre-push.sample
    │   ├── pre-rebase.sample
    │   ├── pre-receive.sample
    │   ├── push-to-checkout.sample
    │   └── update.sample
    ├── info
    │   └── exclude
    ├── objects
    │   ├── info
    │   └── pack
    └── refs
        ├── heads
        └── tags
```
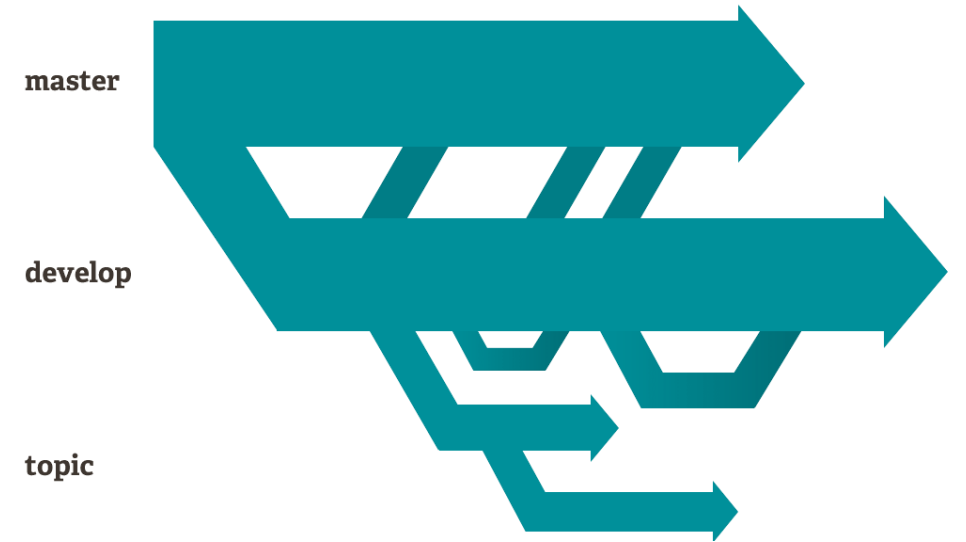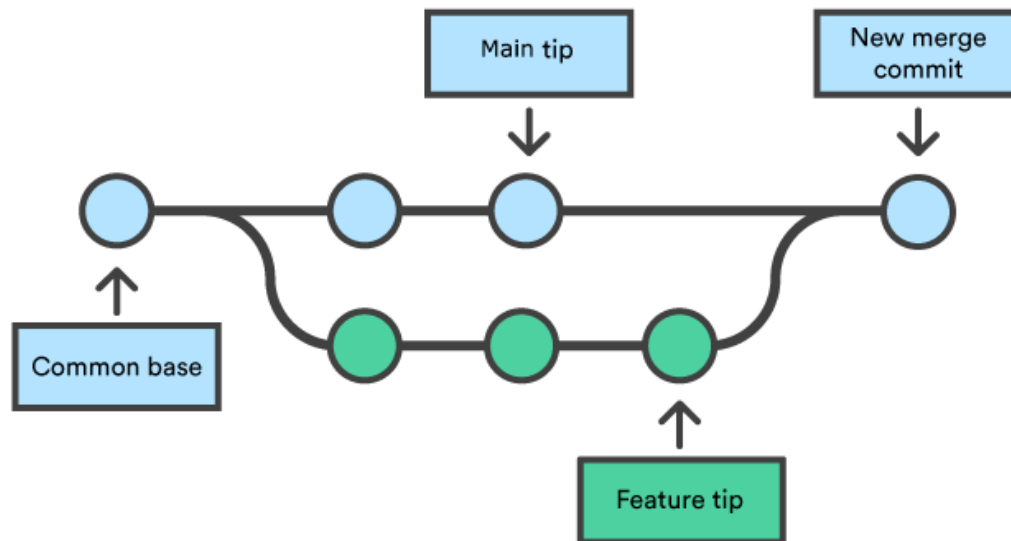
Das git-Repository

- Dateien!
- Wir müssen uns darum nicht kümmern
- Interaktion über CLI oder GUI
- Dateien über .gitignore ignorieren
- „rm –rf .git/" löst hartnäckige Probleme

```
carl@t14s$ git status
Auf Branch master

Noch keine Commits

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

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

Konfiguration

- Minimale Konfiguration:
  Name + Email

- Dotfile:
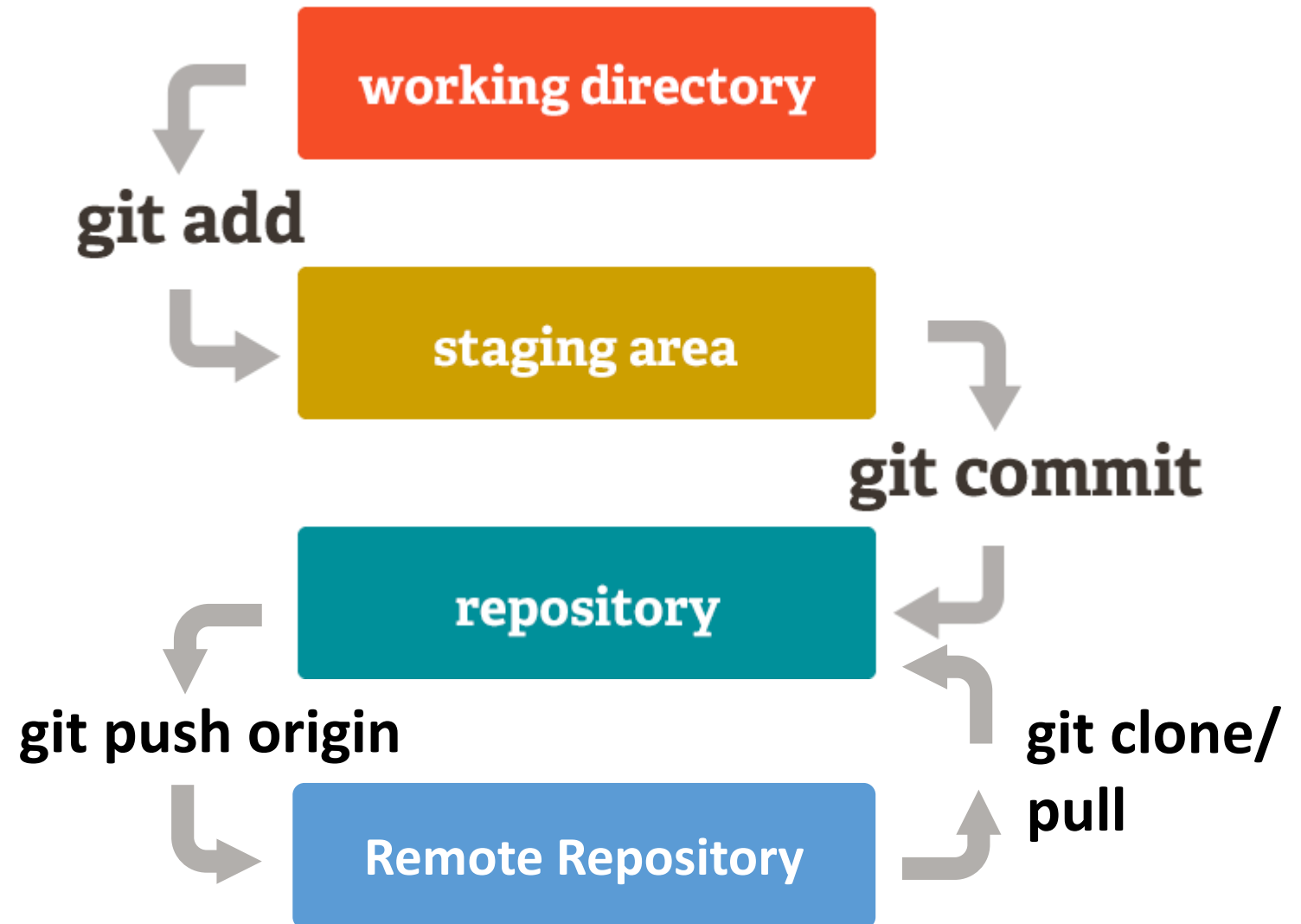 /home/cfbeuchel/.gitconfig

```
[user]
    email = cfbeuchel@imise.uni-leipzig.de
    name = cfbeuchel
    signingKey = 0x197AD7B7ACAB63C6
[filter "lfs"]
    clean = git-lfs clean -- %f
    smudge = git-lfs smudge -- %f
    process = git-lfs filter-process
    required = true

[diff "pandoc"]
    textconv=pandoc --to=markdown
    prompt = false

[alias]
    wdiff = diff --word-diff=color --unified=1
    lg = !"git lg1"
    lg1 = log --graph --abbrev-commit --decorate --format=format:'%C
%C(dim white)- %an%C(reset)%C(bold yellow)%d%C(reset)' --all
    lg2 = log --graph --abbrev-commit --decorate --format=format:'%C
green)(%ar)%C(reset)%C(bold yellow)%d%C(reset)%n'' %C(white)%s%C(res

[color]
    status = auto
    diff = auto
    branch = auto
    interactive = auto
    ui = true
    pager = true
[credential]
    helper = cache
```

Web-based Entwickeln

- Plattform
  - GitHub
  - GitLab
  - SourceForge
  - Launchpad
  - BitBucket
- Self-hosted
  - GitLab
  - Gitea
  - Gogs
  - …

working directory

git add

staging area

git commit

repository

git push origin

git clone/ pull

Remote Repository

Best practices

- Commitumfang - gering
- Commitmessage - ausführlich
- Dokumentation:
  - README.md
  - CHANGELOG
- Tags, Releases
- .gitignore nutzen
- Semantic versioning (https://semver.org/)
- Lizenzen
- (Code of Conduct etc.)
- Tracking von großen Dateien: Git-LFS

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

https://xkcd.com/1296/

Nutze die Doku

- **Niemand merkt sich git Syntax**
- **Git ist komplex**
- GitHub Dokumentation uvm.
- Interaktive Cheatsheets (https://ndpsoftware.com/git-cheatsheet.html)

INSTALLATION & GUIS

With platform specific installers for Git, GitHub also provides the ease of staying up-to-date with the latest releases of the command line tool while providing a graphical user interface for day-to-day interaction, review, and repository synchronization.

GitHub for Windows
https://windows.github.com

GitHub for Mac
https://mac.github.com

For Linux and Solaris platforms, the latest release is available on the official Git web site.

Git for All Platforms
http://git-scm.com

SETUP

Configuring user information used across all local repositories

`git config --global user.name "[firstname lastname]"`

set a name that is identifiable for credit when review version history

`git config --global user.email "[valid-email]"`

set an email address that will be associated with each history marker

`git config --global color.ui auto`

set automatic command line coloring for Git for easy reviewing

SETUP & INIT

Configuring user information, initializing and cloning repositories

`git init`

initialize an existing directory as a Git repository

`git clone [url]`

retrieve an entire repository from a hosted location via URL

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

`git status`

show modified files in working directory, staged for your next commit

`git add [file]`

add a file as it looks now to your next commit (stage)

`git reset [file]`

unstage a file while retaining the changes in working directory

`git diff`

diff of what is changed but not staged

`git diff --staged`

diff of what is staged but not yet committed

`git commit -m "[descriptive message]"`

commit your staged content as a new commit snapshot

BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

`git branch`

list your branches. a * will appear next to the currently active branch

`git branch [branch-name]`

create a new branch at the current commit

`git checkout`

switch to another branch and check it out into your working directory

`git merge [branch]`

merge the specified branch's history into the current one

`git log`

show all commits in the current branch's history

https://education.github.com/git-cheat-sheet-education.pdf

Wie ist GitHub aufgebaut?

torvalds / **linux** Public

Watch 7.9k ▾ | Fork 41.1k | ★ Star 125k ▾

<> Code | ⑂ Pull requests 319 | ⏵ Actions | ▦ Projects | ⚲ Security | ⩘ Insights

⑂ master ▾ | ⑂ 1 branch ⬡ 729 tags

Go to file | Add file ▾ | Code ▾

torvalds Merge tag 'ata-5.17-rc2' of git://git.kernel.org/pub/scm/linux/kernel... ··· 145d9b4 2 hours ago ⏱ **1,072,340** commits

Merge tag 'ata-5.17-rc2' of git://git.kernel.org/pub/scm/linux/kernel...

…/git/dlemoal/libata

```
Pull ATA fix from Damien Le Moal:
 "A single fix for 5.17-rc2, adding a missing resource allocation error
  check in the pata_platform driver, from Zhou"

* tag 'ata-5.17-rc2' of git://git.kernel.org/pub/scm/linux/kernel/git/dlemoal/libata:
  ata: pata_platform: Fix a NULL pointer dereference in __pata_platform_probe()
```

| 📁 | Documentation | Merge tag 'net-5.17-rc2' of git://git.kernel.org/pub/scm/linux/kernel... | 17 hours ago |
📁	LICENSES	LICENSES/LGPL-2.1: Add LGPL-2.1-or-later as valid identifiers	last month
📁	arch	Merge tag 'for-linus' of git://git.armlinux.org.uk/~rmk/linux-arm	3 days ago
📁	block	Merge tag 'bitmap-5.17-rc1' of git://github.com/norov/linux	5 days ago
📁	certs	certs: Fix build error when CONFIG_MODULE_SIG_KEY is empty	6 days ago
📁	crypto	lib/crypto: add prompts back to crypto libraries	10 days ago
📁	drivers	Merge tag 'ata-5.17-rc2' of git://git.kernel.org/pub/scm/linux/kernel...	2 hours ago
📁	fs	Merge tag 'nfs-for-5.17-1' of git://git.linux-nfs.org/projects/anna/l...	3 days ago
📁	include	Merge tag 'net-5.17-rc2' of git://git.kernel.org/pub/scm/linux/kernel...	17 hours ago
📁	init	lib/stackdepot: allow optional init and stack_table allocation by kvm...	6 days ago
📁	ipc	proc: remove PDE_DATA() completely	6 days ago
📁	kernel	Merge tag 'powerpc-5.17-2' of git://git.kernel.org/pub/scm/linux/kern...	5 days ago
📁	lib	Merge tag 'bitmap-5.17-rc1' of git://github.com/norov/linux	5 days ago

About

Linux kernel source tree

📖 Readme
⚖ View license
★ 125k stars
👁 7.9k watching
⑂ 41.1k forks

Releases

⬡ **729** tags

Packages

No packages published

Contributors 5,000+

+ 12,201 contributors

Languages

● C 98.4% ● Assembly 0.9%
● Shell 0.3% ● Makefile 0.2%

Ein einfacher Workflow

- Online: https://rogerdudler.github.io/git-guide/index.de.html
- (Fork repository -> in der Regel haben Sie keine Schreibrechte im Original)
- Clone Repository
- Neuen Branch erstellen
- Etwas ändern
- Änderung commiten
- Commits pushen
- Gepushte Commits mergen
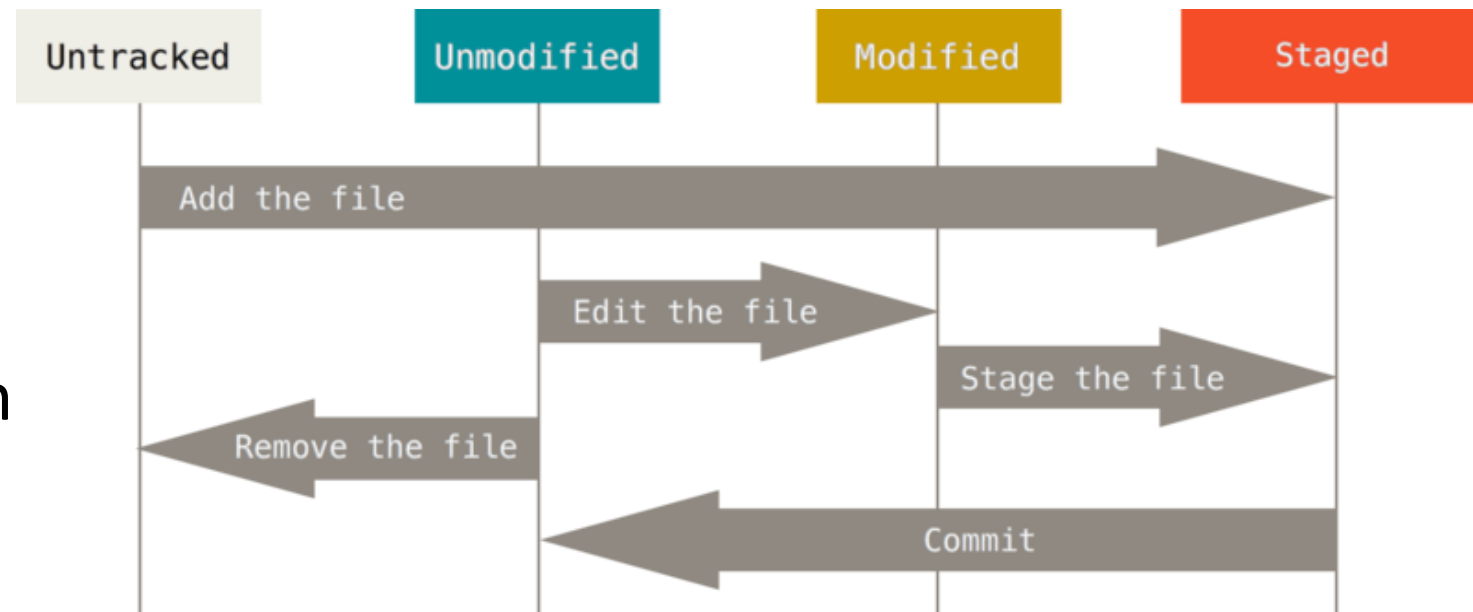
Ein einfacher Workflow

- Fork Repository: https://github.com/GenStatLeipzig/Teaching_StatAspekte

- Clone Repository: https://github.com/"DEIN_ACCOUNT"/Teaching_StatAspekte

- Neuen Branch erstellen - z.B. exercise_r_1_beuchel

- Etwas ändern - .Rmd mit Übungsaufgabe kopieren und lösen

- Änderung commiten

- Commits pushen

- Gepushte Commits per pull request in Hauptrepository integrieren

Ein einfacher Workflow - Fork

- Eine Kopie des Repository im eigenen Account anlegen
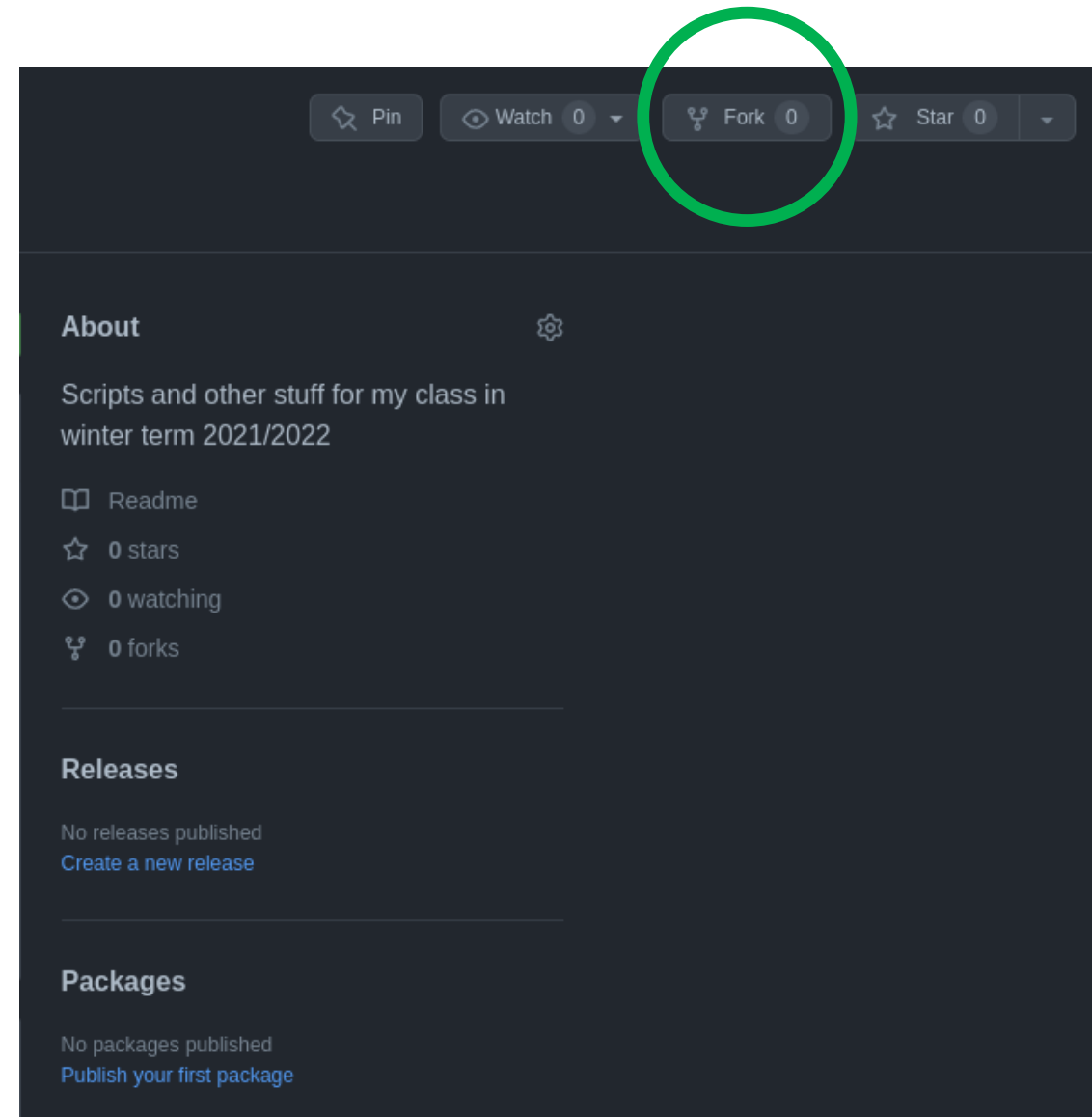
Ein einfacher Workflow - Clone

```
carl@t14s$ cd /tmp/
carl@t14s$ mktemp -d
/tmp/tmp.lO1CNlfSbB
carl@t14s$ cd tmp.lO1CNlfSbB/
carl@t14s$ git clone https://github.com/cfbeuchel/Teaching_StatAspekte
Klone nach 'Teaching_StatAspekte'...
remote: Enumerating objects: 471, done.
remote: Counting objects: 100% (471/471), done.
remote: Compressing objects: 100% (322/322), done.
remote: Total 471 (delta 213), reused 393 (delta 144), pack-reused 0
Empfange Objekte: 100% (471/471), 23.27 MiB | 5.07 MiB/s, fertig.
Löse Unterschiede auf: 100% (213/213), fertig.
carl@t14s$ cd Teaching_StatAspekte/
carl@t14s$ git status
Auf Branch main
Ihr Branch ist auf demselben Stand wie 'origin/main'.

nichts zu committen, Arbeitsverzeichnis unverändert
carl@t14s$ 
```
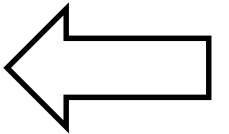
Ein einfacher Workflow - Branch

```
carl@t14s$ git remote -v                                      ← Check Upstream Repositories
origin  https://github.com/cfbeuchel/Teaching_StatAspekte (fetch)
origin  https://github.com/cfbeuchel/Teaching_StatAspekte (push)
carl@t14s$ git branch -a                                      ← Check exitierende Branches
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main                                         ← Erstelle und wechsle
carl@t14s$ git checkout -b uebung_6_loesung_a1                   zu neuem Branch
Zu neuem Branch 'uebung_6_loesung_a1' gewechselt
carl@t14s$ git branch -a
  main
* uebung_6_loesung_a1
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
carl@t14s$ cp Exercises_R/RBlatt6.Rmd Exercises_git/RBlatt6.Rmd   ← Übungsblatt kopieren
carl@t14s$ git add Exercises_git/RBlatt6.Rmd                      ← Neue Kopie tracken
carl@t14s$ git commit -a -m "copy exercise"                       ← Neue Kopie stagen
[uebung_6_loesung_a1 691dbb2] copy exercise                         & commiten
 1 file changed, 231 insertions(+)
 create mode 100644 Exercises_git/RBlatt6.Rmd
carl@t14s$ git status                                         ← Repository Status
Auf Branch uebung_6_loesung_a1
nichts zu committen, Arbeitsverzeichnis unverändert
carl@t14s$
```

Ein einfacher Workflow – Work & Diff

- In Rstudio Exercises_git/RBlatt6.Rmd bearbeiten und abspeichern

```
carl@t14s$ git diff
diff --git a/Exercises_git/RBlatt6.Rmd b/Exercises_git/RBlatt6.Rmd
index 886c85a..3bb5602 100644
--- a/Exercises_git/RBlatt6.Rmd
+++ b/Exercises_git/RBlatt6.Rmd
@@ -115,6 +115,14 @@ a) Bestimmen sie folgende Parameter für alle SNPs *g1* -
 b) Bezogen auf den Standardfehler erster Ordnung, welche genetische Variante
d wo unterscheiden sich die Fehler erster und zweiter Ordnung am meisten?

 c) Unterscheidet sich der kausale Schätzer von der beobachteten Assoziation?
+
+## Lösung Aufgabe 1
+
+a) Es ist doch ganz offensichtlich, dass...
+
+b) Was soll denn diese dumme Frage?
+
```

Ein einfacher Workflow – Stage, Check & Commit

```
carl@t14s$ git status
Auf Branch uebung_6_loesung_a1
Änderungen, die nicht zum Commit vorgemerkt sind:
  (benutzen Sie "git add <Datei>...", um die Änderungen zum Commit vorzumerken)
  (benutzen Sie "git restore <Datei>...", um die Änderungen im Arbeitsverzeichnis zu verwerfen)
        geändert:       Exercises_git/RBlatt6.Rmd

keine Änderungen zum Commit vorgemerkt (benutzen Sie "git add" und/oder "git commit -a")
carl@t14s$ git add Exercises_git/RBlatt6.Rmd
carl@t14s$ git commit Exercises_git/RBlatt6.Rmd -m "Add solution to rblatt6 exercise 1"
[uebung_6_loesung_a1 8c70566] Add solution to rblatt6 exercise 1
 1 file changed, 9 insertions(+), 1 deletion(-)
carl@t14s$ git status
Auf Branch uebung_6_loesung_a1
nichts zu committen, Arbeitsverzeichnis unverändert
carl@t14s$
```

Ein einfacher Workflow – Push

```
carl@t14s$ git remote -v
origin  https://github.com/cfbeuchel/Teaching_StatAspekte (fetch)
origin  https://github.com/cfbeuchel/Teaching_StatAspekte (push)
carl@t14s$ git branch -v
  main
* uebung_6_loesung_a1
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
carl@t14s$ git push origin uebung_6_loesung_a1
Username for 'https://github.com': cfbeuchel
Password for 'https://cfbeuchel@github.com':
Objekte aufzählen: 10, fertig.
Zähle Objekte: 100% (10/10), fertig.
Delta-Kompression verwendet bis zu 16 Threads.
Komprimiere Objekte: 100% (8/8), fertig.
Schreibe Objekte: 100% (8/8), 4.82 KiB | 4.82 MiB/s, fertig.
Gesamt 8 (Delta 3), Wiederverwendet 0 (Delta 0), Pack wiederverwendet 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
remote:
remote: Create a pull request for 'uebung_6_loesung_a1' on GitHub by visiting:
remote:       https://github.com/cfbeuchel/Teaching_StatAspekte/pull/new/uebung_6_loesung_a1
remote:
To https://github.com/cfbeuchel/Teaching_StatAspekte
 * [new branch]      uebung_6_loesung_a1 -> uebung_6_loesung_a1
carl@t14s$ 
```

Ein einfacher Workflow – Merge to Main via pull request

1.)

2.)
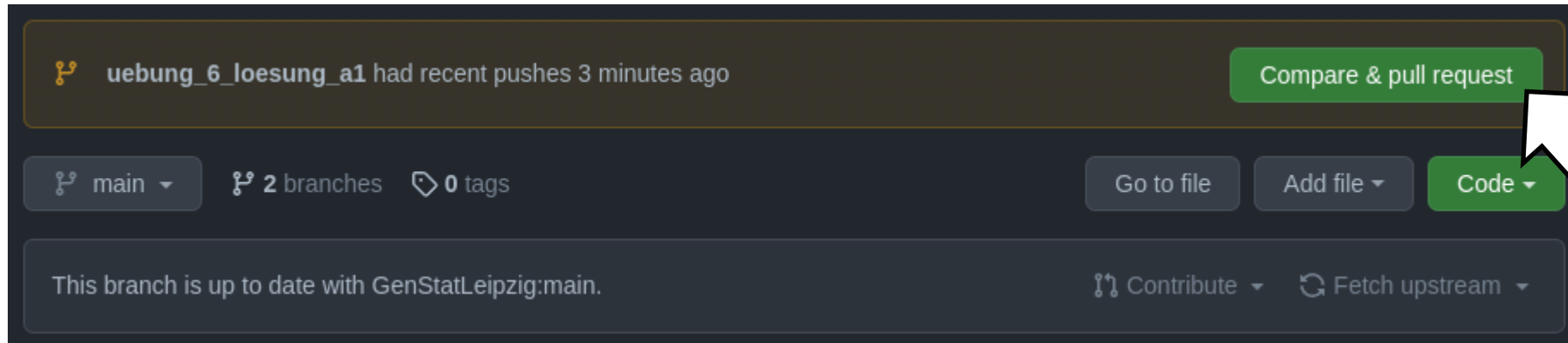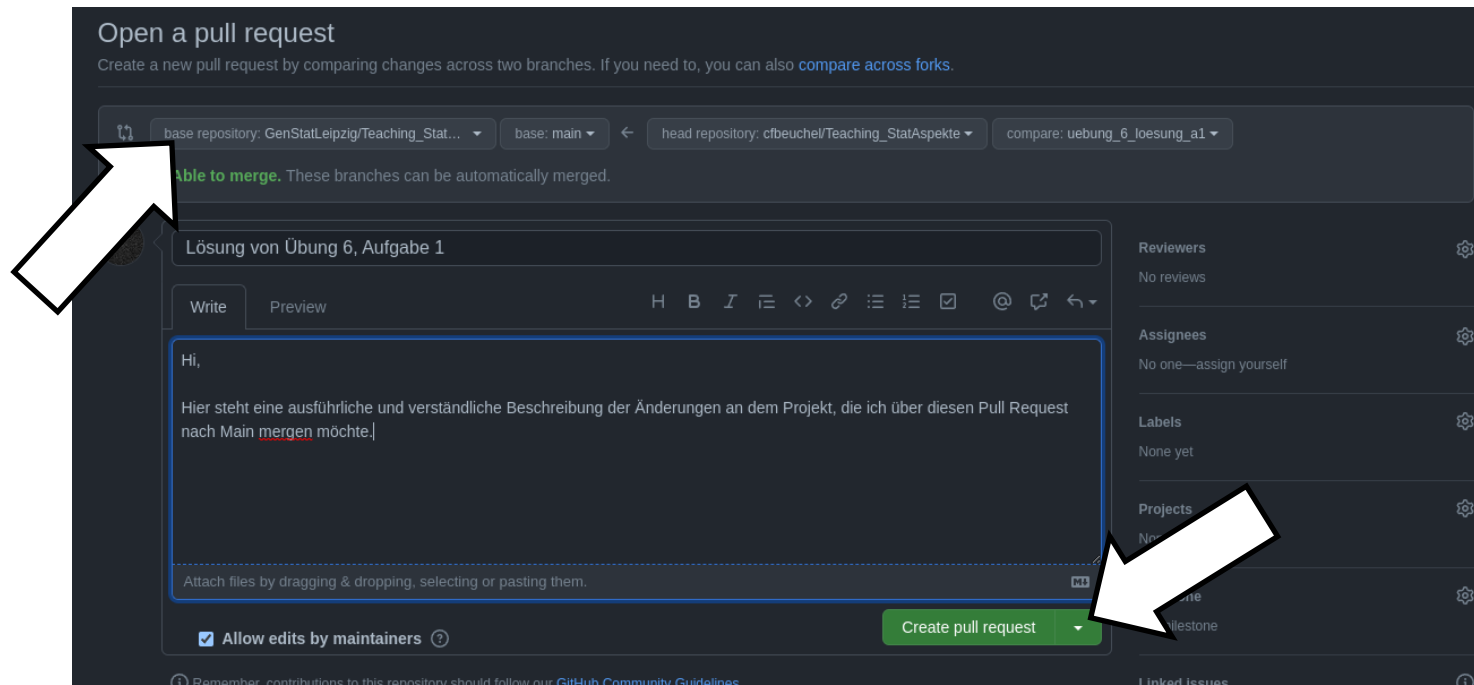
Merge!

<> Code   ⊙ Issues   ⨎ Pull requests 1   ⊳ Actions   ⊞ Projects   ☐ Wiki   ⊙ Security   ⬚ Insights   ⚙ Settings

Lösung von Übung 6, Aufgabe 1 #1

⨎ Open   cfbeuchel wants to merge 2 commits into `GenStatLeipzig:main` from `cfbeuchel:uebung_6_loesung_a1` ⧉

💬 Conversation 0   ⊸ Commits 2   ✓ Checks 0   ⊡ Files changed 1

cfbeuchel commented now   Member ☺ ⋯

Hi,

Hier steht eine ausführliche und verständliche Beschreibung der Änderungen an dem Projekt, die ich über diesen Pull Request nach Main mergen möchte.

cfbeuchel added 2 commits 37 minutes ago

⊸ ⊗ copy exercise   691dbb2

⊸ ⊗ Add solution to rblatt6 exercise 1   8c70566

Add more commits by pushing to the **uebung_6_loesung_a1** branch on **cfbeuchel/Teaching_StatAspekte**.

🤖 **Continuous integration has not been set up**
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

✓ **This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request ▾   or view command line instructions.

Frage:
Was wäre Merge-Konflikt?