

Tools

Git and (other) Tools for Cooperation

Jörg Cassens, Jens Rademacher, Bastian Stender

SoSe 2018

Lab Course Media Informatics



medieninformatik

IMAI – Institut für
Mathematik und
Angewandte Informatik

Outline

git: Theory

Architectures

git

git: Use

git: Tools

Projekt

References

- 1 git: Theory
 - Architectures
 - git

- 2 git: Use

- 3 git: Tools

- 4 Projektmanagement

Use of and Requirements for Version Control

git: Theory

Architectures

git

git: Use

git: Tools

Projekt

References

- Administer different versions of a file

Use of and Requirements for Version Control

git: Theory

Architectures

git

git: Use

git: Tools

Projekt

References

- Administer different versions of a file
- Log of changes
 - What,
 - When,
 - Who
- Possible to use previous versions
- Multi-user support
- Support branching, merging, redundancy

git: Theory

Architectures

git

git: Use

git: Tools

Projekt

References

Architectures

Architectures (1)

git: Theory

Architectures

git

git: Use

git: Tools

Projekt

References

- Local version control
 - Versioning single files with simple administration (log, recover older versions of file)
 - **Implementations:** RCS, proprietary software
- First generation, not suitable for groups

Architectures (2)

git: Theory

Architectures

git

git: Use

git: Tools

Projekt

References

- Central version control
 - Central server, development on clients
 - Revision history on server
 - Rights management on server
 - **Implementations:** CVS (abandoned), SVN
- Second generation, suitable for groups, needs server

Architectures (3)

git: Theory

Architectures

git

git: Use

git: Tools

Projekt

References

- Distributed version control
 - Every client has a repository
 - No central server necessary
 - but primus inter pares possible
 - Repositories update other repositories
 - Version history might be on every client
 - Parallel development with (tool-supported) merge afterwards (non-linear development)
 - **Implementations:** Mercurial, Git
- Third generation, suitable for groups, supports offline and non-linear development

Outline

git: Theory

Architectures

git

git: Use

git: Tools

Projekt

References

git

- Distributed version control
- Originally developed by Linus Torvalds for the Linux kernel
- no central server
- Supporting non-linear development through branching and merging
- Lots of transport protocol options
- No incremental ID, but hash-values for commits
- Authentication with repository-hosting services (primus inter pares) often via private keys

(Dis-) advantages

git: Theory

Architectures

git

git: Use

git: Tools

Projekt

References

■ Advantages

- no central server
- clean file system, only one “.git” directory in main directory
- efficient work through branch, diff, merge

■ Disadvantages

- Requires discipline
- Linux thinking, might be difficult coming from Windows
- Steep learning curve
- Slow performance with large (and binary) files

Outline

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References

1 git: Theory

2 git: Use

- New Repository
- Standard Tasks
- Remote-Repositories
- Branching
- Other
- Tutorial

3 git: Tools

4 Projektmanagement

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References

New Repository

Initialisation & .gitignore

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

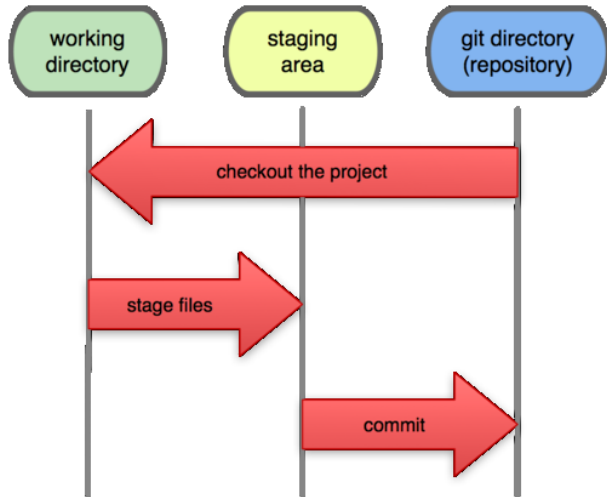
git: Tools

Projekt

References

- New repositories:
 - **git init** (local in project directory, creates “.git”)
 - **git --bare init** (remote repository)
- Keeping files or file types out of version control:
 - text file **.gitignore** in main directory
 - One rule per line (***.pdf**)

Local Operations



Workflow

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References

Standard Tasks

Base Functions

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

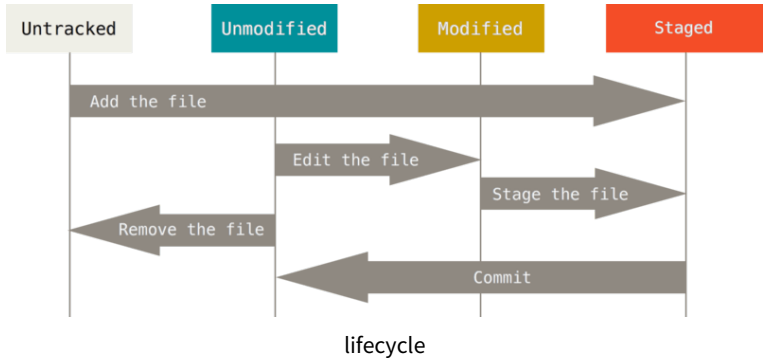
Tutorial

git: Tools

Projekt

References

- File-Status:
 - **Untracked** (not under version control)
 - **Modified** (changed since last commit)
 - **Staged** (marked for commit)
 - **Unmodified** (unchanged since last commit)
- Read status:
 - **git status**
- Add or stage files:
 - **git add FILE(S)**
- Commit:
 - **git commit -m "Comment"**
- Stage (deleted & modified) & Commit:
 - **git commit -a -m "Comment"**
- Show differences to HEAD:
 - **git diff HEAD**



Revert changes

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References

- Revert changes of last commit, keep commit
 - **git revert**
- HEAD-pointer to named commit
 - **git revert COMMIT**
- Stage/working directory to status of last commit
 - **git reset**

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References

Remote-Repositories

Remote-Repositories

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References

- make local copy of remote repository:
 - **git clone URL**
- add remote repository:
 - **git remote add REMOTENAME URL**
- transfer local changes to remote repository:
 - **git push REMOTENAME BRANCH**
 - REMOTENAME is often **origin**
 - Standard-Branch is **master**

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References

Branching

Branching

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References

- create branch
 - **git branch NAME**
- show all branches
 - **git branch -a**
- change to branch
 - **git checkout NAME**
- merge branch with master
 - **git checkout master** (change into master)
 - **git merge NAME** (Merge)
- delete branch
 - **git branch -d NAME**

Branches

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References



working with branches

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References

Other

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References

- (Global) settings
 - **git config [--global] user.name “John Doe“**
 - **git config [--global] user.email john@example.com**
 - **git config [--global] core.autocrlf input (Linux)**
- Help
 - **git COMMAND --help**

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

Other

Tutorial

git: Tools

Projekt

References

Tutorial

github Tutorial

git: Theory

git: Use

New Repository

Standard Tasks

Remote-Repositories

Branching

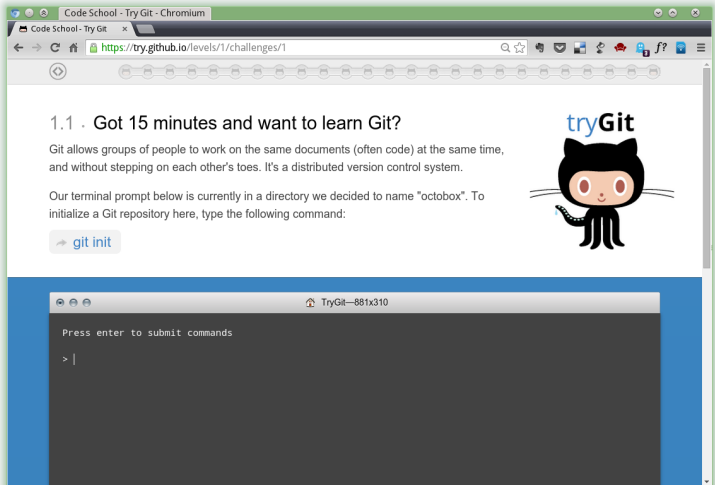
Other

Tutorial

git: Tools

Projekt

References



 try.github.com

Outline

git: Theory

git: Use

git: Tools

git: GUI

git: Project Hosting

Projekt

References

1 git: Theory

2 git: Use

3 git: Tools

- git: GUI

- git: Project Hosting

4 Projektmanagement

git: GUI

git: Theory

git: Use

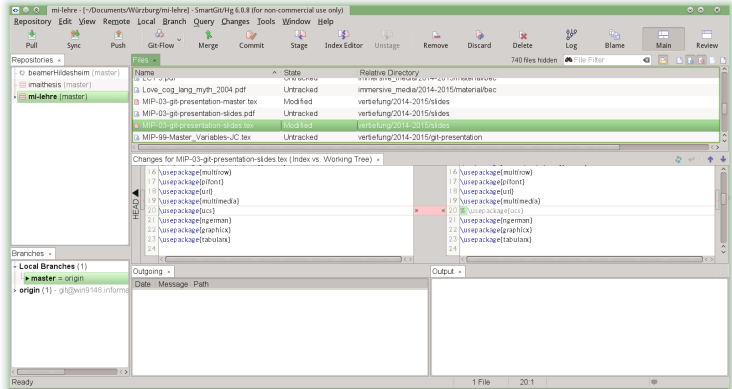
git: Tools

git: GUI

git: Project Hosting

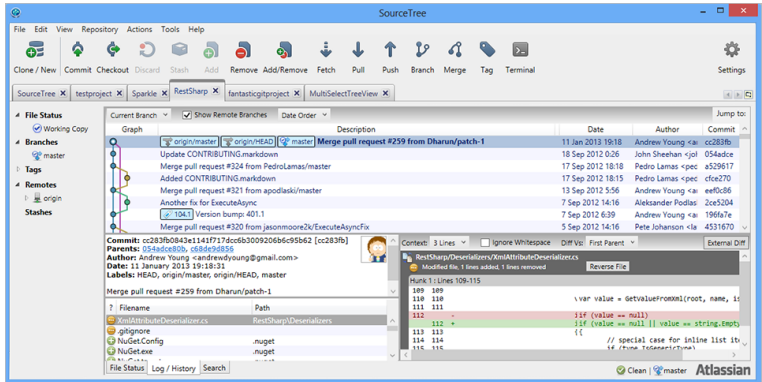
Projekt

References



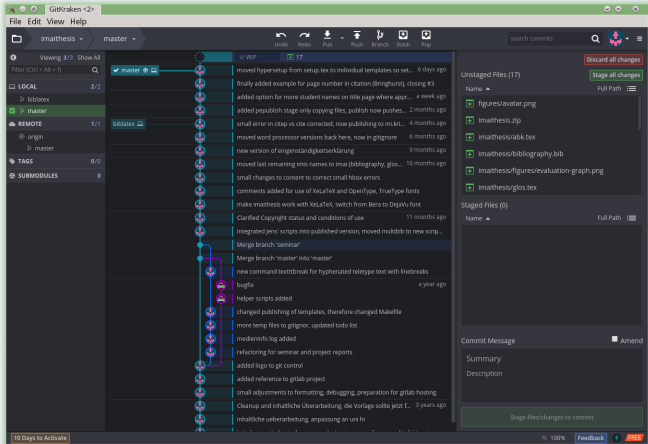
🖱️ www.syntevo.com/smartgit

Cross-platform (Linux, Mac, Windows), free for non-commercial use



SourceTreeApp.com

Mac, Windows; free to use, registration required



🖱️ gitkraken.com

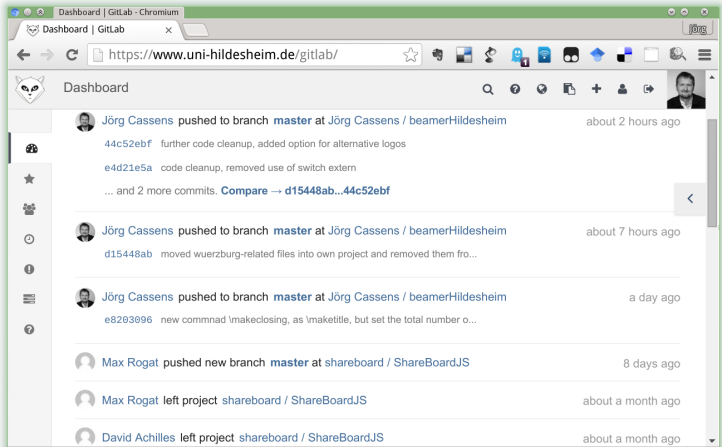
Cross-platform (Linux, Mac, Windows), free for non-commercial use,
registration required

git: Project Hosting

- Help for projects by offering:
 - Issue tracker
 - Wiki (Markdown)
 - Statistics (Gamification)
 - Download of projects
 - Releases
- Enabling teamwork
- Making forks and pull-requests simple
 - Easy to get involved
 - “Standing on the shoulders of giants”
- Several Services with different (dis-) advantages
 - gitlab
 - github
 - bitbucket

gitlab (Uni Hildesheim)

git: Theory
git: Use
git: Tools
git: GUI
git: Project Hosting
Projekt
References

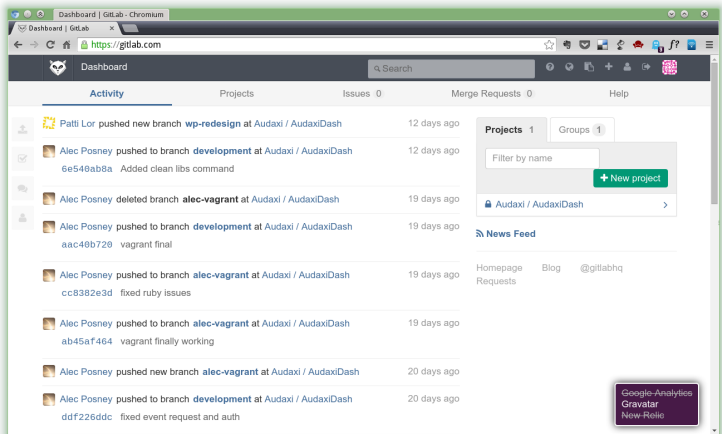


www.uni-hildesheim.de/gitlab
Hosted OSS-System

gitlab (Commercial)

git: Theory
git: Use
git: Tools
git: GUI
git: Project Hosting

Projekt
References



 gitlab.com

Commercial, hosted

gitlab (OSS)

git: Theory

git: Use

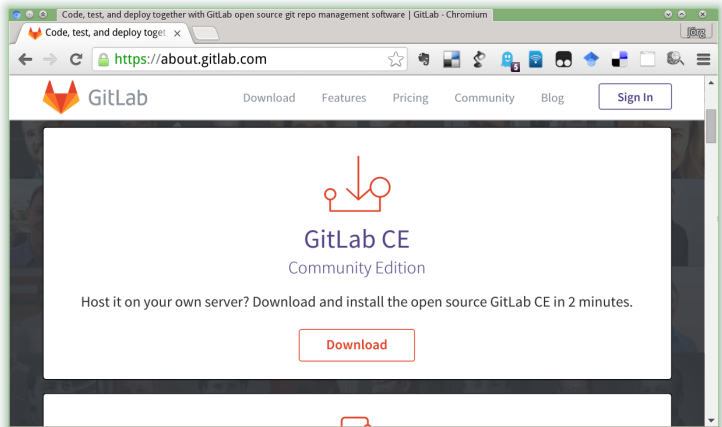
git: Tools

git: GUI

git: Project Hosting

Projekt

References



 about.gitlab.com

Self-hosted OSS-System

Atlassian Bitbucket

git: Theory

git: Use

git: Tools

git: GUI

git: Project Hosting

Projekt

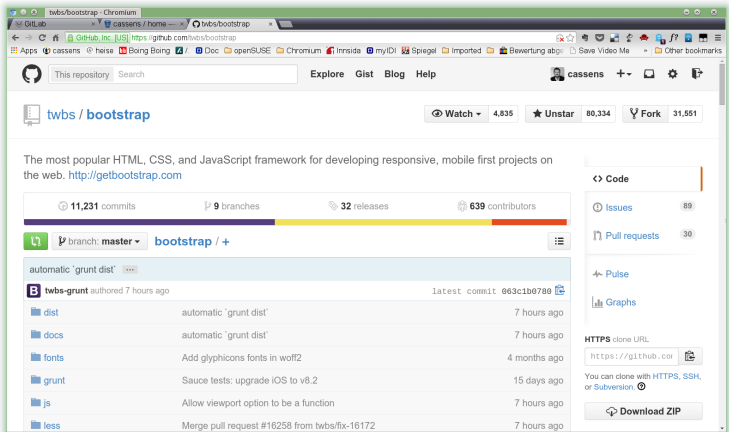
References

The screenshot shows the Bitbucket web interface for the repository 'cassens / beamerHildesheim'. The interface is in German and includes the following elements:

- Overview Section:**
 - Last updated: 2014-10-20
 - Language: LaTeX
 - Access level: Admin
 - 1 Branch
 - 0 Tags
 - 0 Forks
 - 1 Watcher
- Invite users to this repo:** A button labeled 'Send Invitation' and an icon for inviting users.
- Recent activity:**
 - Umstellung auf Würzburger Medieninfo...** (Issue #1 created in cassens/beamerhildesheim) by Jörg Cassens - 2014-11-13
 - 2 commits** (Pushed to cassens/beamerhildesheim):
 - 328d497 fbXcolors option should now work
 - 02c8a81 cleanup and prep for mi at wuerz...
 - 1 commit** (Pushed to cassens/beamerhildesheim):
 - 3ba00ab added a modified theme for medi...
- README Section:** A message stating 'THERE ISN'T A README YET' with a 'Create a README' button.

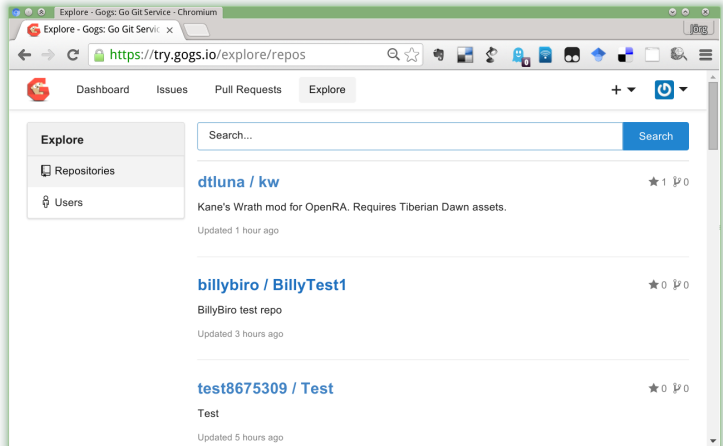
bitbucket.org

Commercial, hosted, Freemium



👉 www.github.com

Commercial, hosted, Freemium



 gogs.io

Self-hosted OSS-System

Use in Lab Course

git: Theory

git: Use

git: Tools

git: GUI

git: Project Hosting

Projekt

References

- Every group should have at least one project on one of the following services
 -  uni-hildesheim.de/gitlab – Universität Hildesheim
 -  www.gitlab.com – free public and private projects
 -  www.bitbucket.com – free public and private projects, limited team size
 -  www.github.com – free public projects
- I get invited
 - Access to code and documentation
 - Issue tickets
- Recommend uni gitlab
- Others are fine as well
 - At least those where I have an account

Outline

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

Communication &
Coordination

Automation

Suggestions

References

1 git: Theory

2 git: Use

3 git: Tools

4 **Projektmanagement**

- Ticketing & Project Planing
- Documentation
- Communication & Coordination
- Automation
- Suggestions

Need

- Version control
 - As seen
- Ticketing
 - Basic version included with hosted services
- Project planing
 - At least: milestones and ticketing
- Documentation
 - Wiki at hosted services
 - $\text{\LaTeX}$ in git
 - Collaborative editors
- Communication & Coordination
 - More than whatsapp, facebook and Dropbox
- Automation
 - When other things happen in the repo, other stuff can be triggered (mail, chat, test)

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

Communication &
Coordination

Automation

Suggestions

References

git: Theory

git: Use

git: Tools

Projekt

**Ticketing & Project
Planing**

Documentation

Communication &
Coordination

Automation

Suggestions

References

Ticketing & Project Planing

Tickets and Milestones

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

Communication &
Coordination

Automation

Suggestions

References

- git-hosted services usually come with ticketing
- With tickets and milestones, rudimentary project planing is already possible
 - Tasks
 - Responsibilities
 - Time
- Advantages
 - Using existing tools & same toolchain
- Disadvantages
 - Not very flexible

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

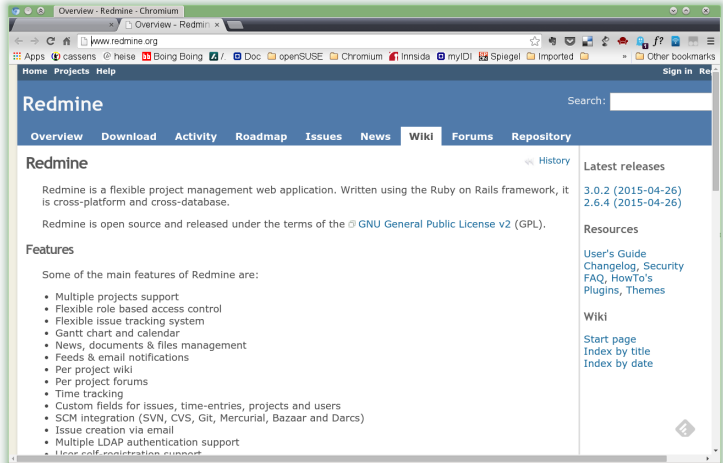
Documentation

Communication &
Coordination

Automation

Suggestions

References



 www.redmine.org

Integrated system

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planning

Documentation

Communication &
Coordination

Automation

Suggestions

References

- Powerful integrated system
 - Project management
 - Time keeping
 - Documentation
 - git-Repositories
- Advantages
 - Lots of options, expandable
- Disadvantages
 - Not a pure project planning solution
 - Self-Hosting
- Alternative
 -  trac

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planning

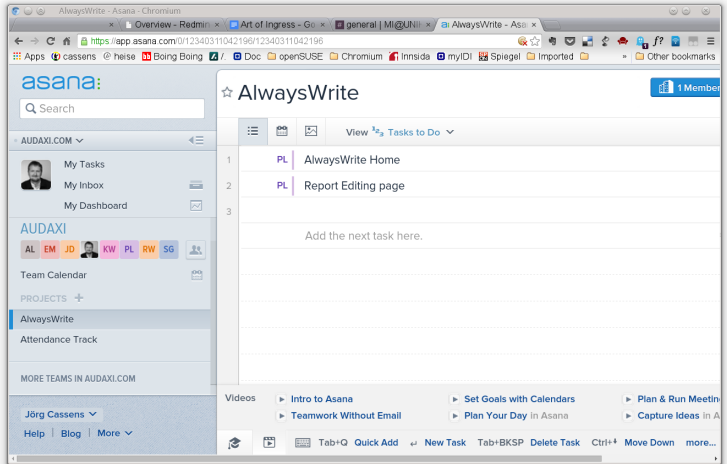
Documentation

Communication &
Coordination

Automation

Suggestions

References



 asana.com

Task management, freemium

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planning

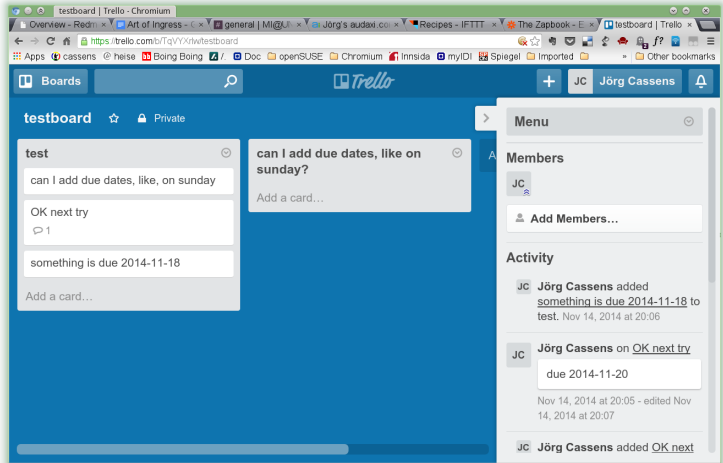
Documentation

Communication &
Coordination

Automation

Suggestions

References



 trello.com

Kanban-style task management, freemium

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planning

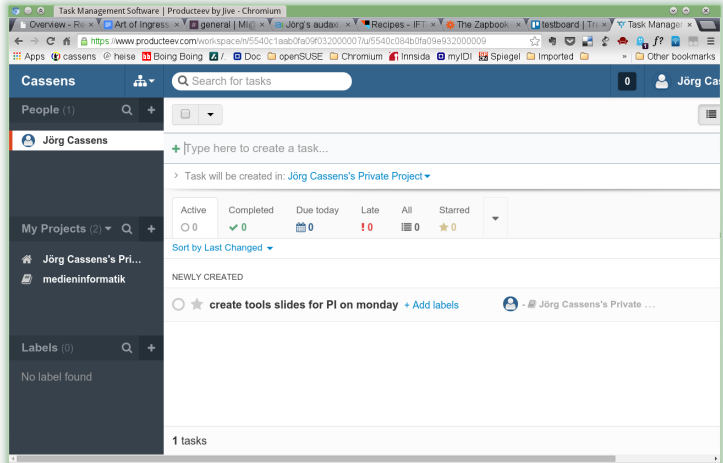
Documentation

Communication &
Coordination

Automation

Suggestions

References



 www.producteev.com

Task management, freemium

asana, Trello, producteev

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planning

Documentation

Communication &
Coordination

Automation

Suggestions

References

- Focus on project management
- Different philosophies (kanban, “traditional”)
- Advantages
 - Powerful
- Disadvantages
 - Powerful

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

Communication &
Coordination

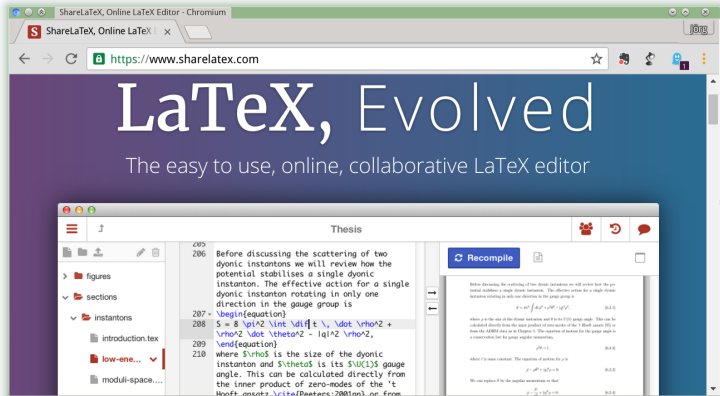
Automation

Suggestions

References

Documentation

- git works well with text files
 - but is not very well suited for binary blobs such as PDF or word processor files
- LaTeX is text based
 - Documentation in a doc-repo
- Disadvantages
 - Assets such as pictures are not handled well
 - Track changes
 - needs work with  latexdiff
- Advantages
 - Professional type setting
 - Same tool chain



👉 sharelatex.com

Collaborative, online $\text{\LaTeX}$ -shell, Freemium

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

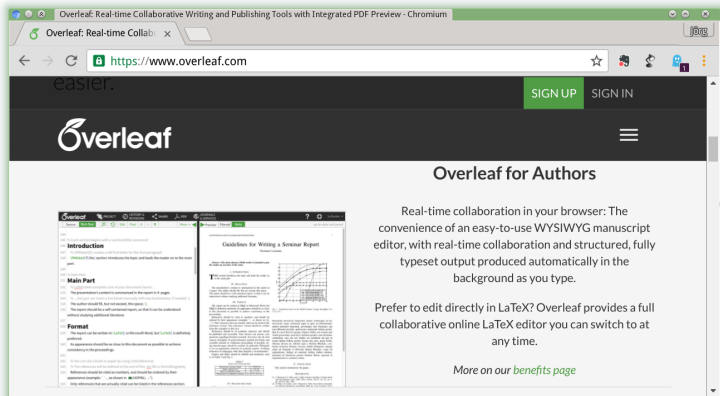
Documentation

Communication &
Coordination

Automation

Suggestions

References



🖱️ [overleaf.com](https://www.overleaf.com)

Collaborative, online $\text{\LaTeX}$ -shell, Freemium

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

Communication &
Coordination

Automation

Suggestions

References

- “pre-packaged” Wikis
- Advantages
 - Simple Markdown-Syntax
 - Online
 - Same tool chain
- Disadvantages
 - Simple Markdown-Syntax
 - Online
- You can convert Markdown into lots of other formats (HTML, PDF, $\text{\LaTeX}$, ODF)
 - Example pandoc,  johnmacfarlane.net/pandoc
 - Example MultiMarkdown,
 fletcherpenney.net/multimarkdown

Etherpad

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

Communication &
Coordination

Automation

Suggestions

References

The screenshot shows the Etherpad web interface. At the top is a toolbar with various editing tools: bold (B), italic (I), underline (U), strikethrough (ABC), bulleted list, numbered list, indent, outdent, undo, redo, and a color picker. To the right of the toolbar are icons for share, clock, star, settings, code view, and a user icon showing 1 user. The main editing area contains the following text:

```

1  ### header
2  Welcome to Etherpad Lite!
3  This pad text is synchronized as you type, so that everyone viewing this page sees
   the same text. This allows you to collaborate seamlessly on documents!
4
5  Etherpad Lite on Github: http://i.mp/ep-lite
6
7  Hum is anyone there? Yes, There is! Well this looks good. anyone else here?
8  Wow this is a really nice pad!
9  trying the service so, how good is this? seems nice..!
10 lalalalalala This has great influence on people!
11
12 We're massively grateful for your support
13 lalalalla la l l
16 as this is no more of any importance,
17 we could acutally remove it.
  
```

On the right side, there is a vertical scrollbar and a chat window at the bottom right with the text "Chat" and a speech bubble icon.

etherpad.org, epad.hosting.uni-hildesheim.de
OSS Collaborative online text editor, hosted or self-hosted

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planning

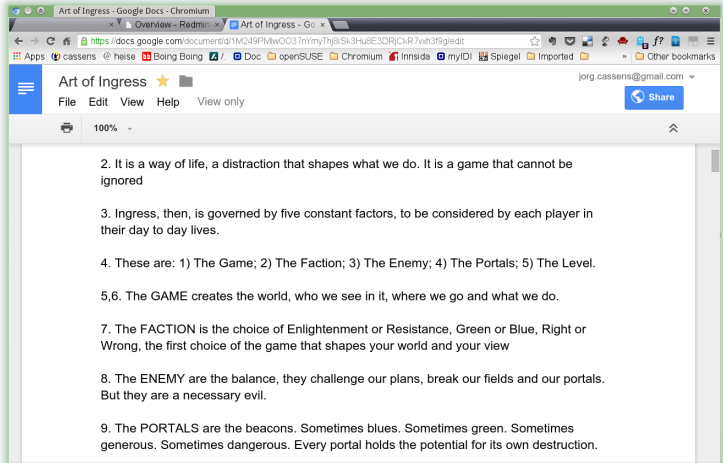
Documentation

Communication &
Coordination

Automation

Suggestions

References



 docs.google.com

Collaborative online word processor

Microsoft Word Online

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planning

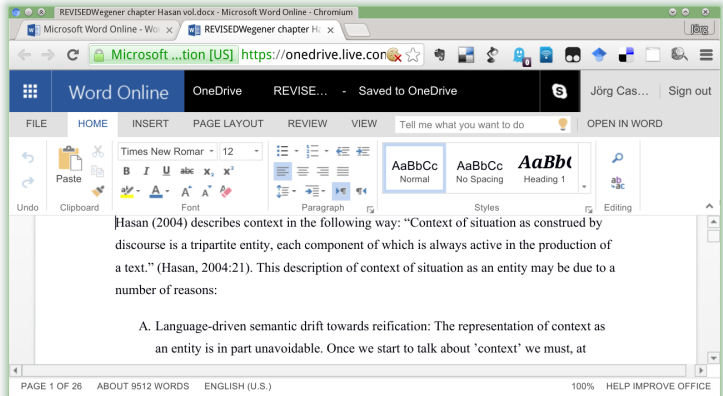
Documentation

Communication &
Coordination

Automation

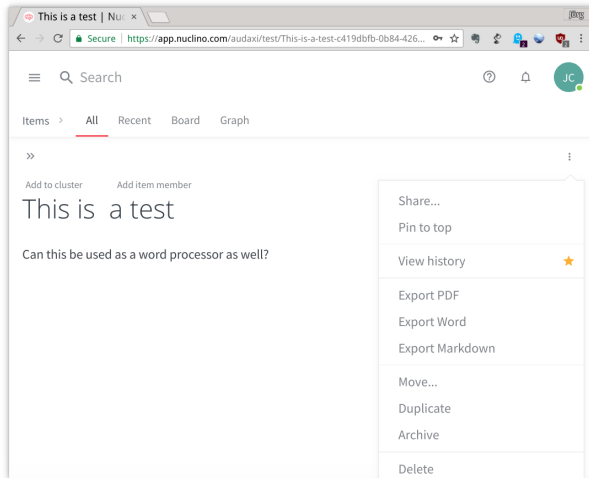
Suggestions

References



office.live.com

Collaborative online word processor



 nuclino.com

“Easy knowledge base for teams”, includes collaborative word processor

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

**Communication &
Coordination**

Automation

Suggestions

References

Communication & Coordination

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

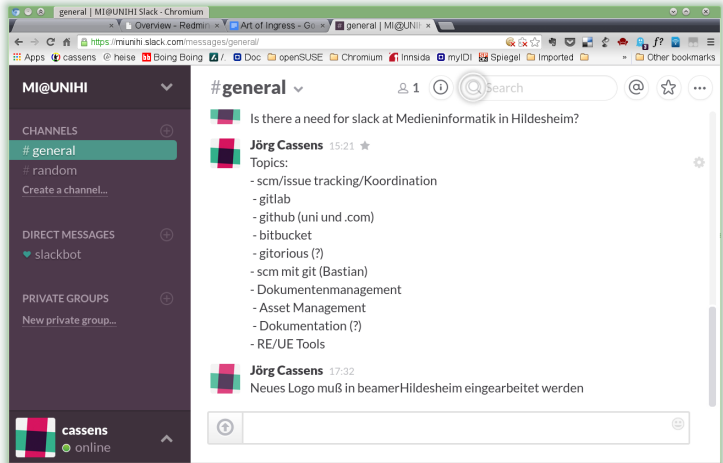
Documentation

Communication &
Coordination

Automation

Suggestions

References



 slack.com

Commercial, Freemium

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

Communication &
Coordination

Automation

Suggestions

References

- Communication in Teams in so-called channels
- More structured than Hangouts, less messy than facebook, simpler than IRC
- Advantages
 - Many functions and hooks
 - Good connectivity with other systems (git commit-messages)
 - Even free accounts quite powerful
- Disadvantages
 - ...

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planning

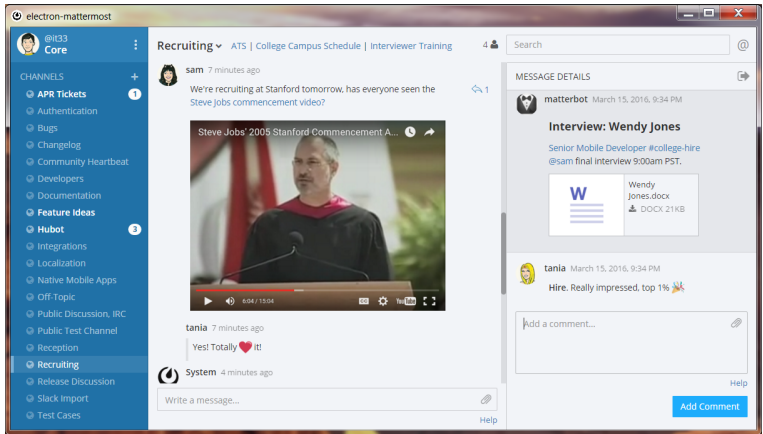
Documentation

Communication &
Coordination

Automation

Suggestions

References



🖱️ mattermost.org

Self-hosted Slack Clone, bei gitlab “dabei”

Telegram

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

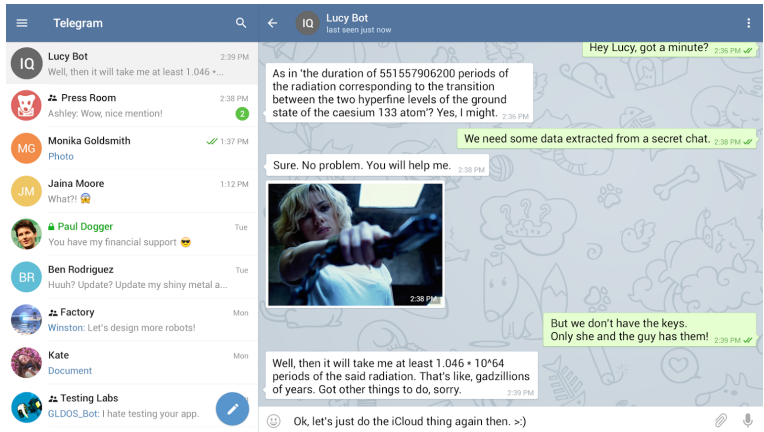
Documentation

Communication &
Coordination

Automation

Suggestions

References



telegram.org

Messenger, End-to-End-Encryption bei privaten chats, Gruppen

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

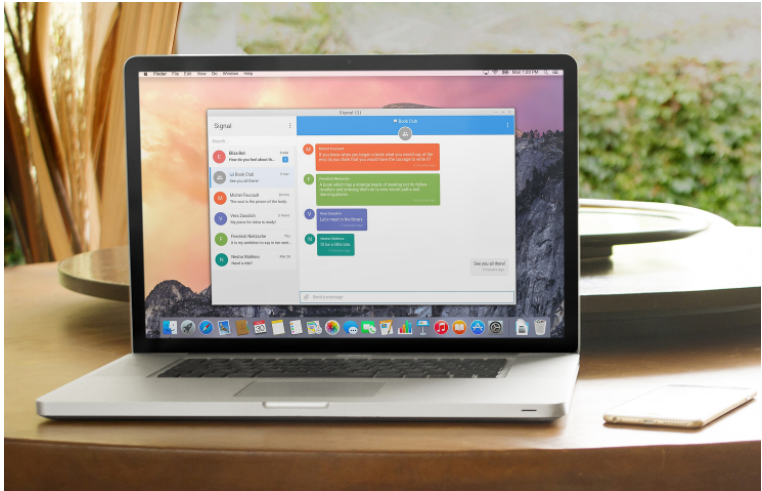
Documentation

Communication &
Coordination

Automation

Suggestions

References



 whispersystems.org

Messenger, End-to-End-Encryption, Desktop-Version, Gruppen

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

Communication &
Coordination

Automation

Suggestions

References

DFN Terminplaner+

Expert-Mode » Frontpage » Praktikum Medieninformatik SoSe 2016

[Edit Foodle](#) [Preferences](#) [Logout](#)

[Open in spreadsheet](#)

[Deutsch](#) | [English](#)

Praktikum Medieninformatik SoSe 2016

Termine in der Vorlesungszeit für Treffen aller Teilnehmenden (nicht nur einzelner Projektgruppen).

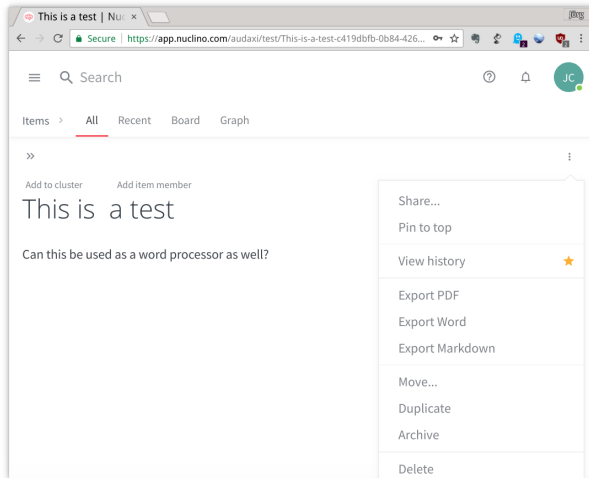
This foodle is closed.
Unfortunately you cannot update entries or add new one.

My response | All responses | Discussion (0 entries) | Distribute | Delete

Name	Mittwochs			Donnerstags			Bei anderem Termin		Updated
	12:00-14:00	14:00-16:00	16:00-18:00	12:00-14:00	14:00-16:00	16:00-18:00	Kein Interesse	Habe Interesse	
	✓	✓	✓	✗	✗	✗	✗	✓	
	✗	✗	✓	✗	✗	✗	✗	✗	

terminplaner2.dfn.de

Terminplaner, DFN



 nuclino.com

“Easy knowledge base for teams”, includes collaborative word processor

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

Communication &
Coordination

Automation

Suggestions

References

Automation

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

Communication &
Coordination

Automation

Suggestions

References

- Git has the ability to fire custom scripts when certain actions occur
- There are both client-side and server-side hooks
- Most hosted services offer convenient access to such hooks
 - Mail at commit
 - Slack-Messages at commit

if this then that

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planning

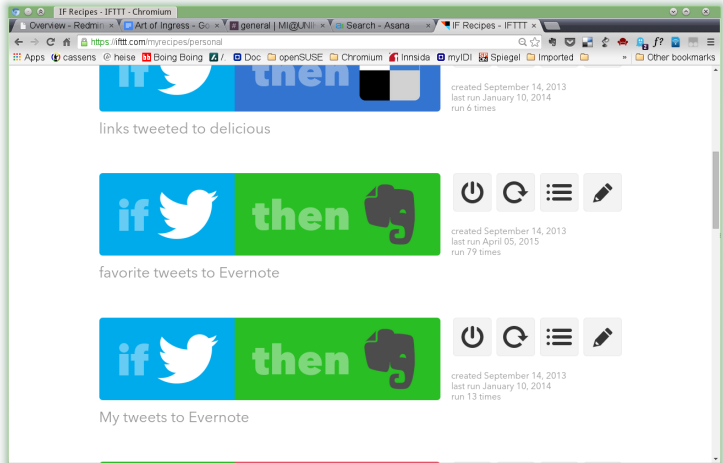
Documentation

Communication &
Coordination

Automation

Suggestions

References



 ifttt.com

If this then that, commercial, freemium

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

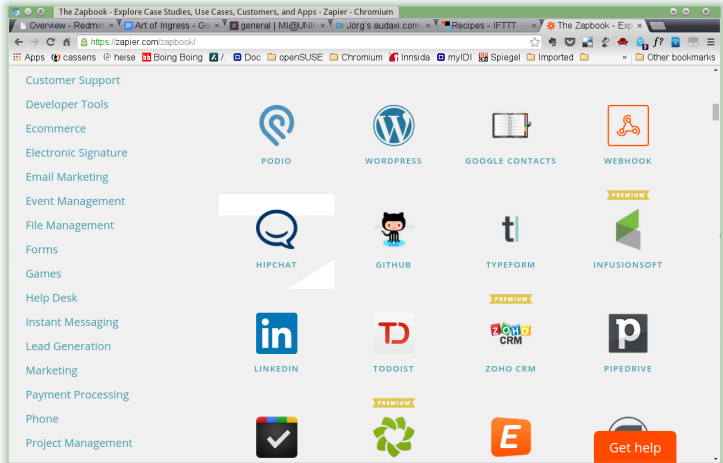
Documentation

Communication &
Coordination

Automation

Suggestions

References



zapier.com

Similar to ifttt, commercial, freemium

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

Communication &
Coordination

Automation

Suggestions

References

- Both services make it possible to connect different data sources and data sinks from different services
- ifttt is more open in how to use stuff, but zapier sometimes has more of better connections
- Advantages
 - Automation
- Disadvantages
 - Ones gives third party services access accounts on a potentially very large number of services...and sometimes to lots of data

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planing

Documentation

Communication &
Coordination

Automation

Suggestions

References

Suggestions

Suggestions

git: Theory

git: Use

git: Tools

Projekt

Ticketing & Project
Planning

Documentation

Communication &
Coordination

Automation

Suggestions

References

- Source-code management (git)
- Git hosted services (gitlab, github, bitbucket)
- Documentation from the start (wiki, $\text{\LaTeX}$)
- Tickets (git hosted services)
- Milestones (git hosted services)
- Chat (slack, Telegram)

- Basis
 -  git-scm.com – Git for Windows (install, deutsche Sprachdatei in .old umbenennen)
- Tutorial & Documentation
 -  try.github.com – Github-tutorial with Octocat
 -  git-scm.com/book – Git book
- GUI-Tools
 -  www.syntevo.com/smartgit
 -  SourceTreeApp.com
 -  gitkraken.com

■ Hosted services

-  uni-hildesheim.de/gitlab – Universität Hildesheim
-  www.gitlab.com – free public and private projects
-  www.bitbucket.com – free public and private projects, limited team size
-  www.github.com – free public projects

■ Self-hosted

-  about.gitlab.com – self-hosted
-  gogs.io – self-hosted

Project Management & Documentation

■ Project management

-  www.redmine.org
-  trac.edgewall.org
-  asana.com
-  trello.com
-  www.producteev.com

■ Documentation

-  $\text{\LaTeX}$ &  [git](http://git-scm.com)
-  sharelatex.com
-  overleaf.com
-  etherpad.org
-  epad.hosting.uni-hildesheim.de
-  docs.google.com
-  office.live.com
-  nuclino.com
-  johnmacfarlane.net/pandoc
-  fletcherpenney.net/multimarkdown

■ Communication & Coordination

-  slack.com – Slack
-  mattermost.org – Slack-Clone
-  telegram.org – telegram
-  whispersystems.org – Signal
-  terminplaner2.dfn.de – Foodle
-  nuclino.com – knowledge management

■ Automation

-  ifttt.com
-  zapier.com

Tools

Git and (other) Tools for Cooperation

Jörg Cassens, Jens Rademacher, Bastian Stender

SoSe 2018

Lab Course Media Informatics



medieninformatik

IMAI – Institut für
Mathematik und
Angewandte Informatik