Tools

Git and (other) Tools for Cooperation

Jörg Cassens, Jens Rademacher, Bastian Stender

Lab Course Media Informatics

Universität Hildesheim
medieninformatik
IMAI – Institut für Mathematik und Angewandte Informatik

Inhaltsverzeichnis

1 git: Theory

Use of and Requirements for Version Control

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

1.1 Architectures

Architectures (1)

- 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)

- 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)

- 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

1.2 git

Features

- 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

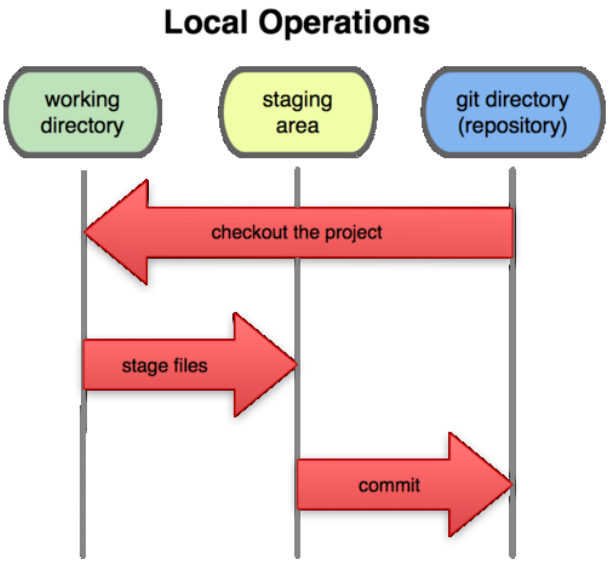
- Advantages
  - no central server
  - clean file system, only one ".git" directory in main directory
  - efficient work through branch, diff, merge
- Disadvantages
  - Requires discipline
  - Linux thinking, might be difficult coming from Windows
  - Steep learning curve
  - Slow performance with large (and binary) files

2 git: Use

2.1 New Repository

Initialisation & .gitignore

- 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**)

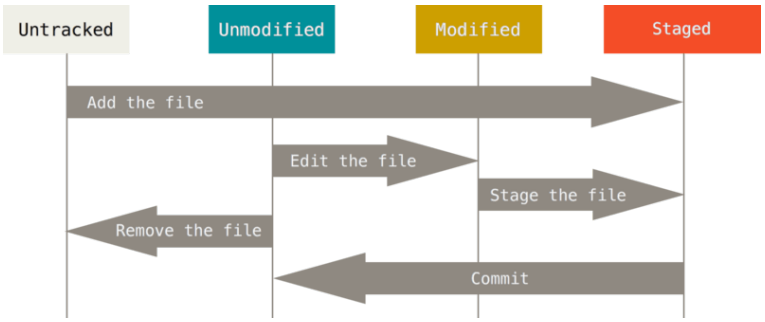


Workflow

2.2 Standard Tasks

Base Functions

- 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**



lifecycle

Revert changes

- 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**

2.3 Remote-Repositories

Remote-Repositories

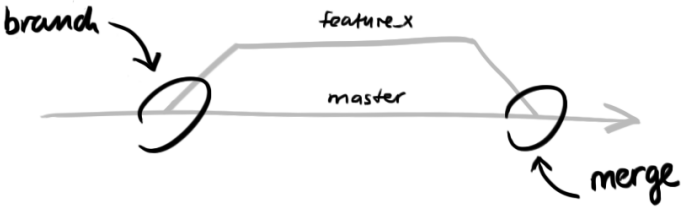
- make local copy of remote repository:
  - **git clone URL**
- add remote repository:
  - **git remote add REMOTENAME URL**
- transfer local changes to remote repository:
  - **git push REMOTENAME BRANCH**
  - REMOTENAME is often **origin**
  - Standard-Branch is **master**

2.4 Branching

Branching

- 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



working with branches

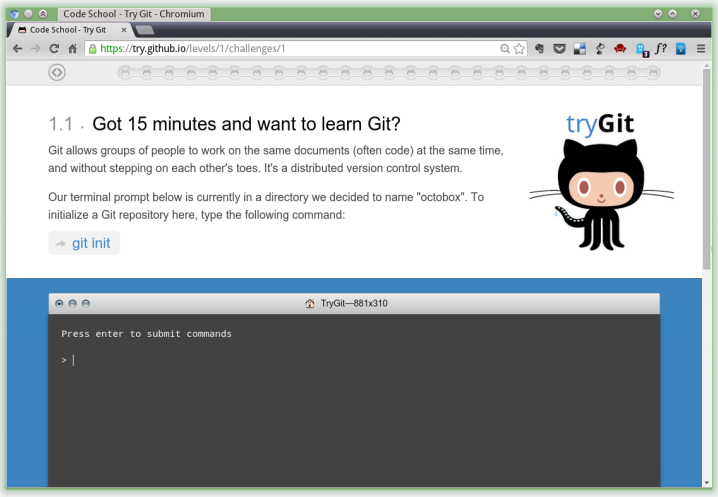
2.5 Other

Setting and help

- (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**

2.6 Tutorial

github Tutorial

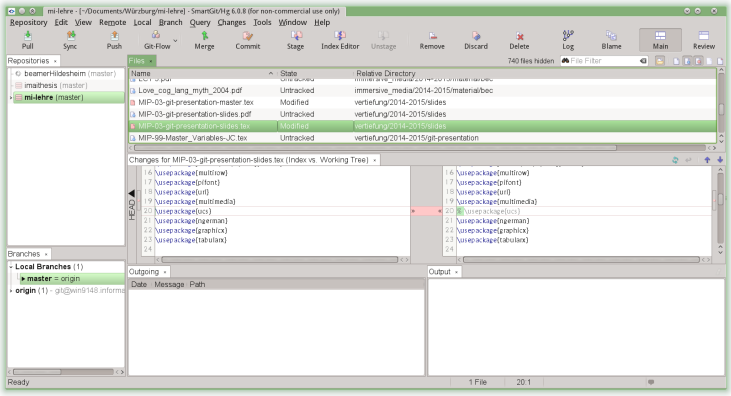


☞ try.github.com

3 git: Tools

3.1 git: GUI

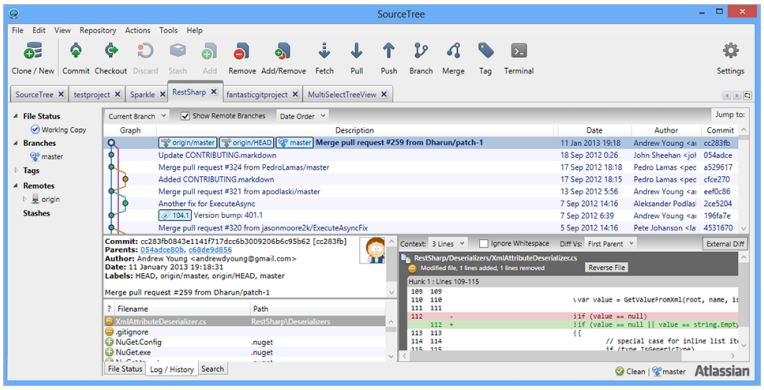
smartgit



☞ www.syntevo.com/smartgit

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

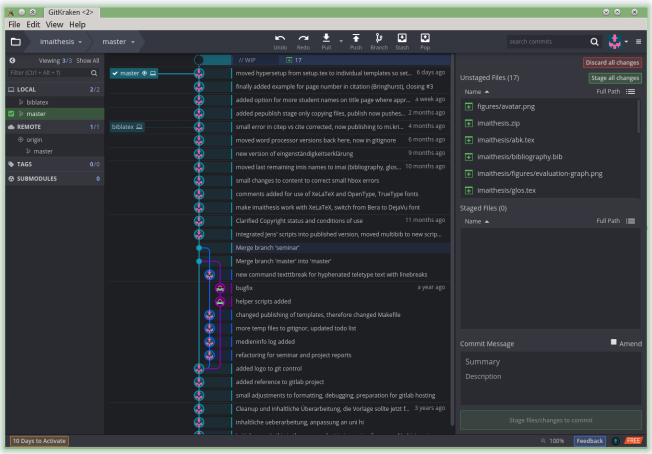
SourceTree



☞ SourceTreeApp.com

Mac, Windows; free to use, registration required

GitKraken



☞ gitkraken.com

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

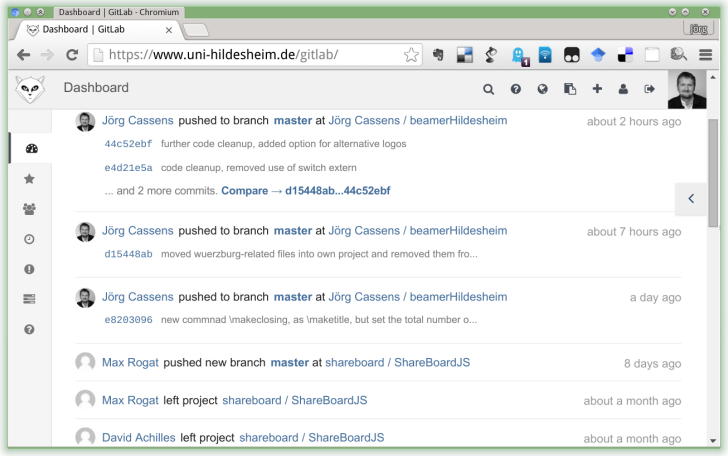
3.2 git: Project Hosting

Project Hosting

- Help for projects by offering:
  - Issue tracker
  - Wiki (Markdown)

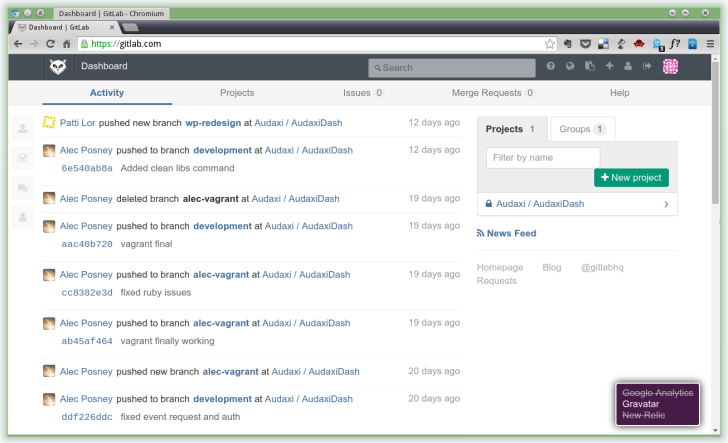
  - Statistics (Gamification)
  - Download of projects
  - Releases
- Enabling teamwork
- Making forks and pull-requests simple
  - Easy to get involved
  - “Standing on the shoulders of giants”
- Several Services with different (dis-) advantages
  - gitlab
  - github
  - bitbucket

gitlab (Uni Hildesheim)



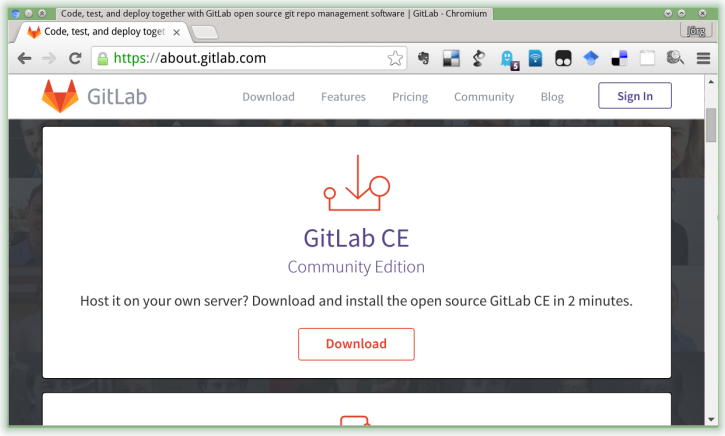
☞ www.uni-hildesheim.de/gitlab
Hosted OSS-System

gitlab (Commercial)



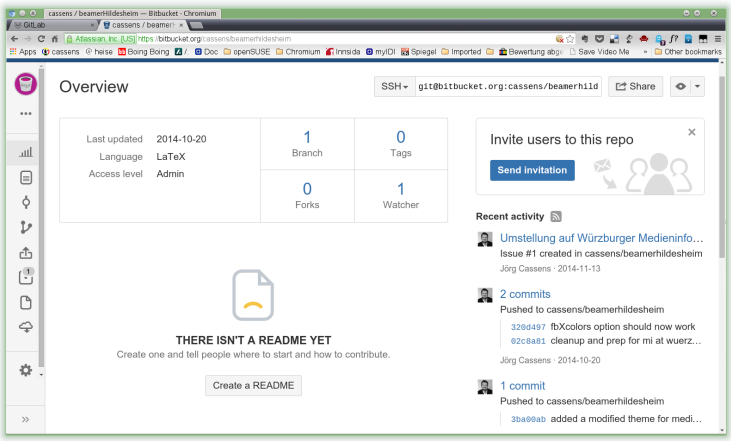
☞ gitlab.com
Commercial, hosted

gitlab (OSS)



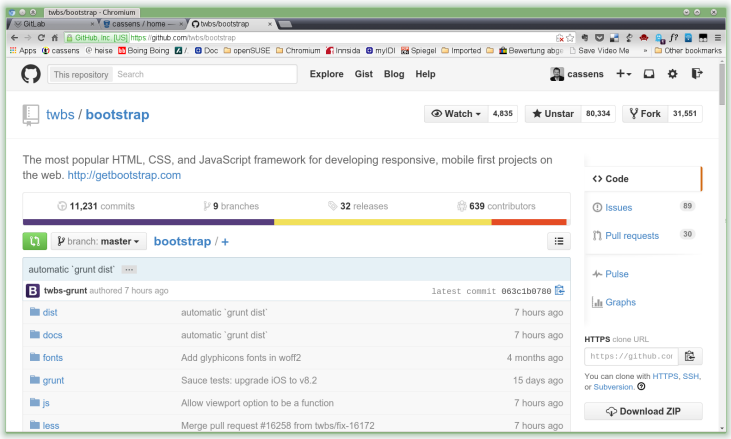
☞ about.gitlab.com
Self-hosted OSS-System

Atlassian Bitbucket



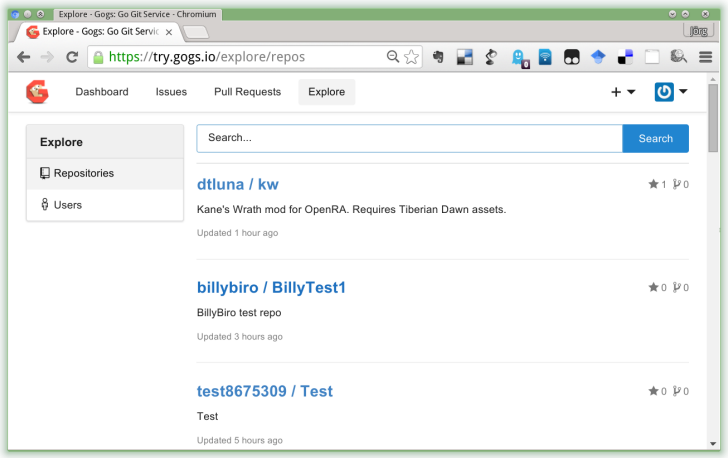
☞ bitbucket.org
Commercial, hosted, Freemium

github



☞ www.github.com
Commercial, hosted, Freemium

gogs



☞ gogs.io
Self-hosted OSS-System

Use in Lab Course

- 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

4 Projektmanagement

Need

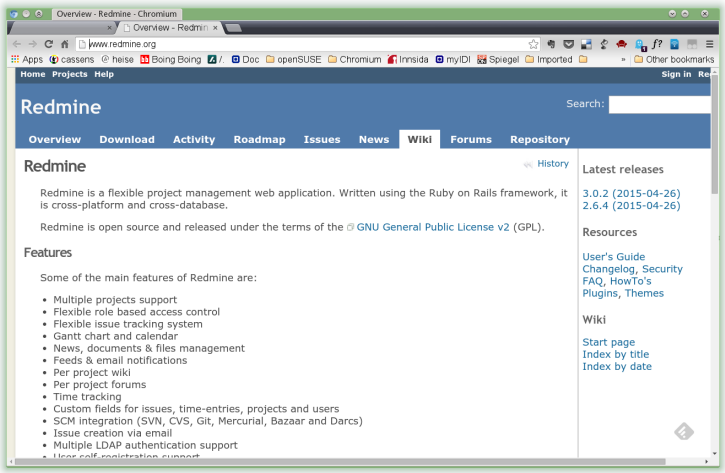
- Version control
  - As seen
- Ticketing
  - Basic version included with hosted services
- Project planing
  - At least: milestones and ticketing
- Documentation
  - Wiki at hosted services
  - LaTeX in git
  - Collaborative editors
- Communication & Coordination
  - More than whatsapp, facebook and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

4.1 Ticketing & Project Planing

Tickets and Milestones

- 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

Redmine

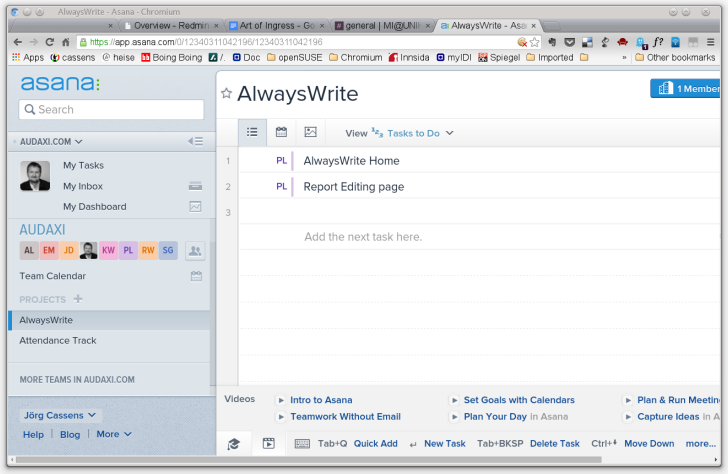


☞ www.redmine.org
Integrated system

Redmine

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

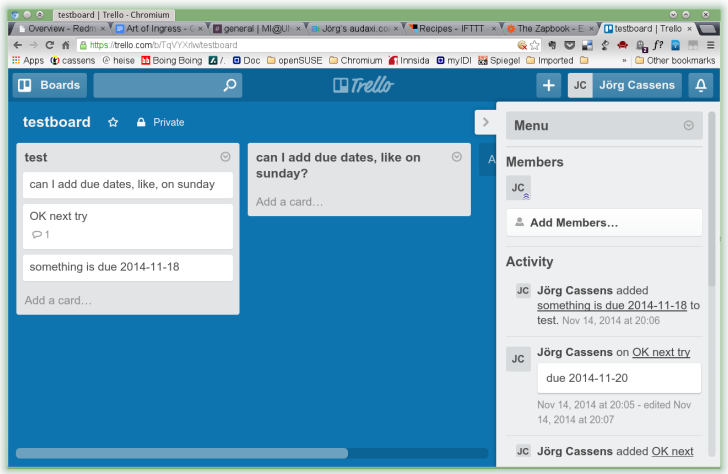
asana



☞ asana.com

Task management, freemium

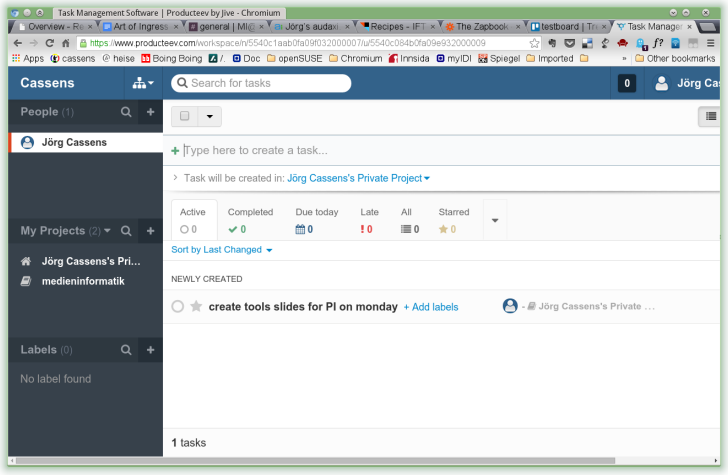
Trello



☞ trello.com

Kanban-style task management, freemium

producteev



☞ www.producteev.com

Task management, freemium

asana, Trello, producteev

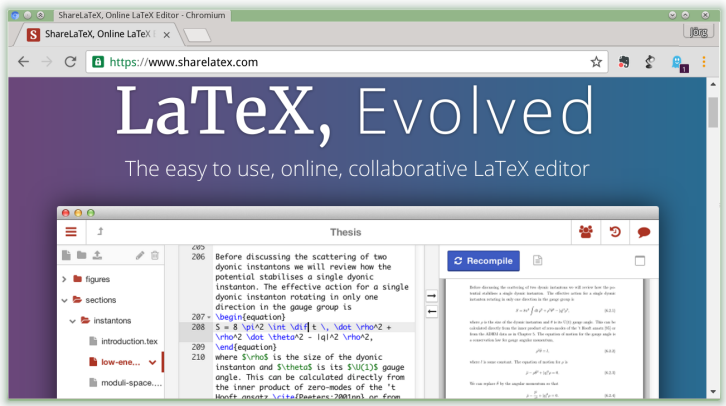
- Focus on project management
- Different philosophies (kanban, “traditional”)
- Advantages
  - Powerful
- Disadvantages
  - Powerful

4.2 Documentation

LaTeX

- 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

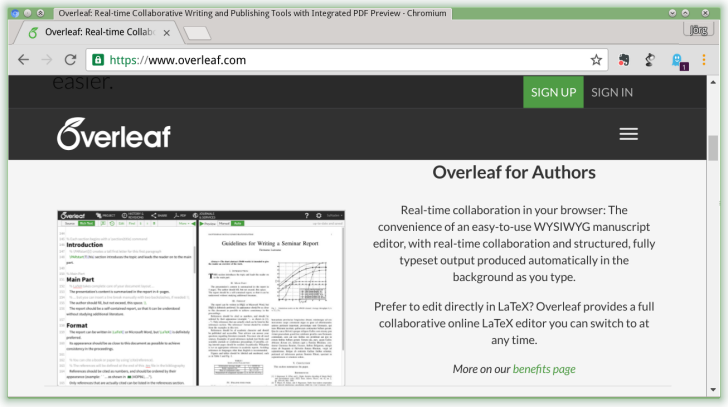
ShareLaTeX



☞ sharelatex.com

Collaborative, online LaTeX-shell, Freemium

overleaf

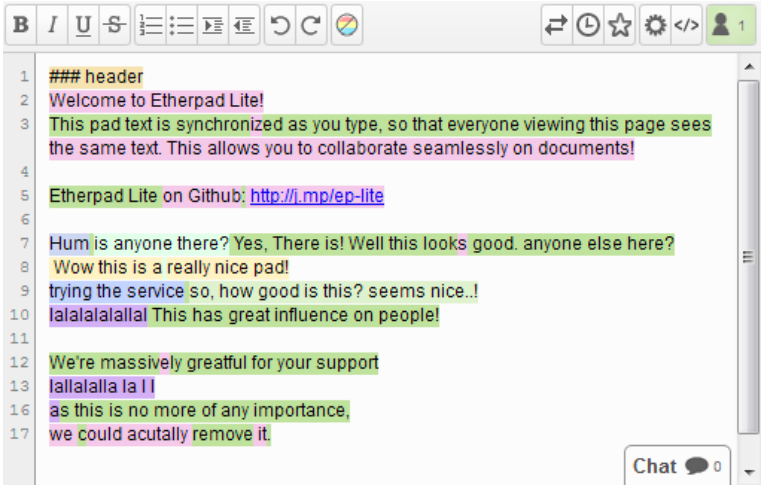


☞ overleaf.com
Collaborative, online LaTeX-shell, Freemium

Wiki

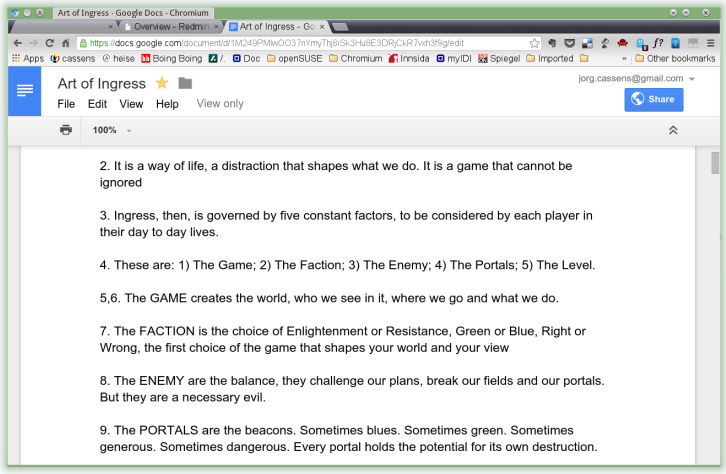
- "pre-packaged" Wikis
- Advantages
  - Simple Markdown-Syntax
  - Online
  - Same tool chain
- Disadvantages
  - Simple Markdown-Syntax
  - Online
- You can convert Markdown into lots of other formats (HTML, PDF, LaTeX, ODF)
  - Example pandoc, ☞ johnmacfarlane.net/pandoc
  - Example MultiMarkdown,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

Etherpad



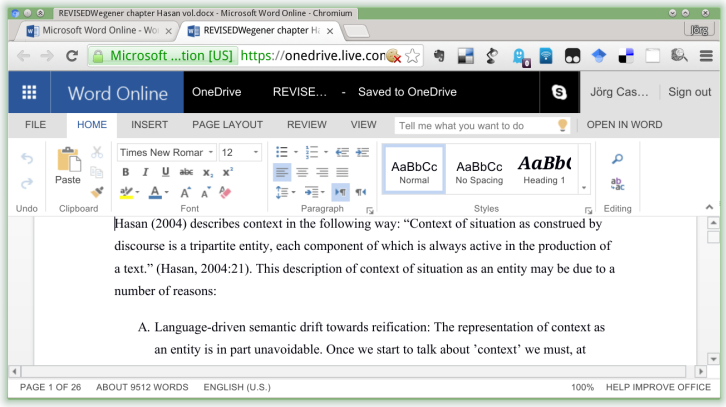
☞ etherpad.org, ☞ epad.hosting.uni-hildesheim.de
OSS Collaborative online text editor, hosted or self-hosted

Google Docs



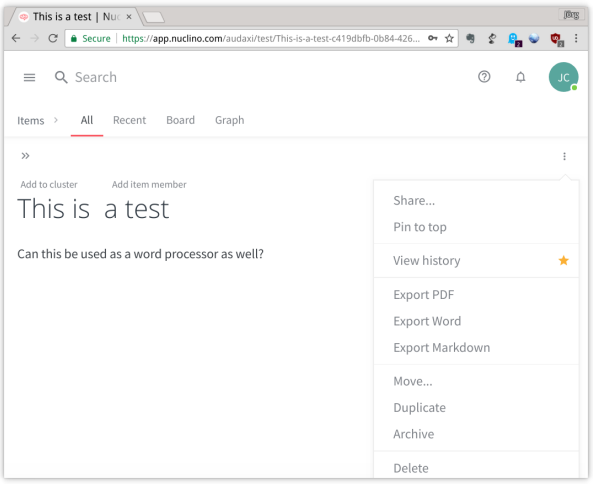
☞ docs.google.com
Collaborative online word processor

Microsoft Word Online



☞ office.live.com
Collaborative online word processor

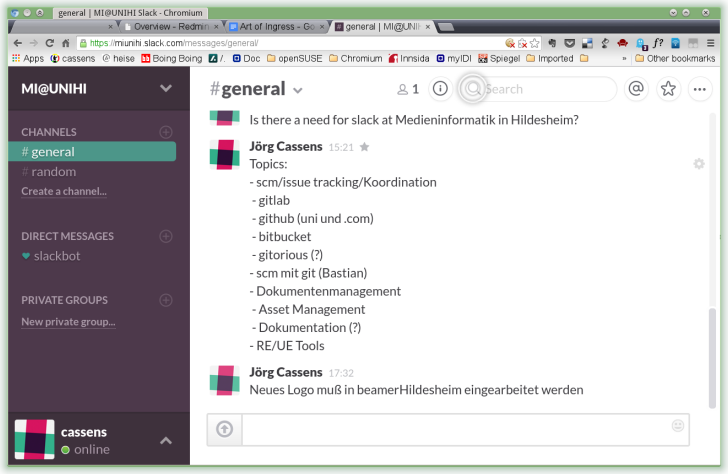
Nuclino



☞ nuclino.com
"Easy knowledge base for teams", includes collaborative word processor

4.3 Communication & Coordination

slack

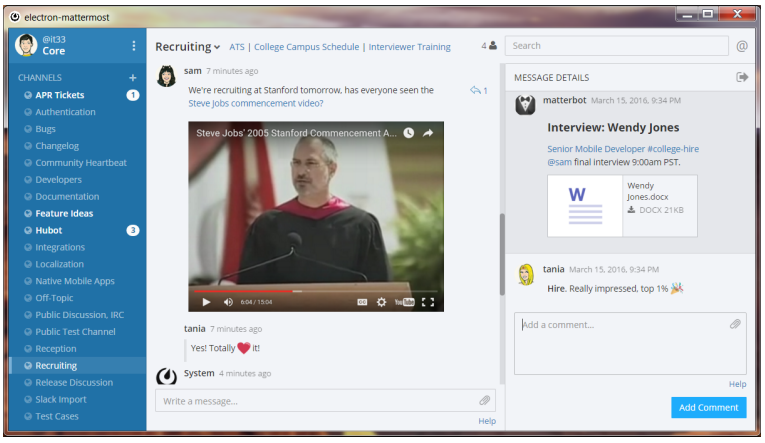


☞ slack.com
Commercial, Freemium

slack

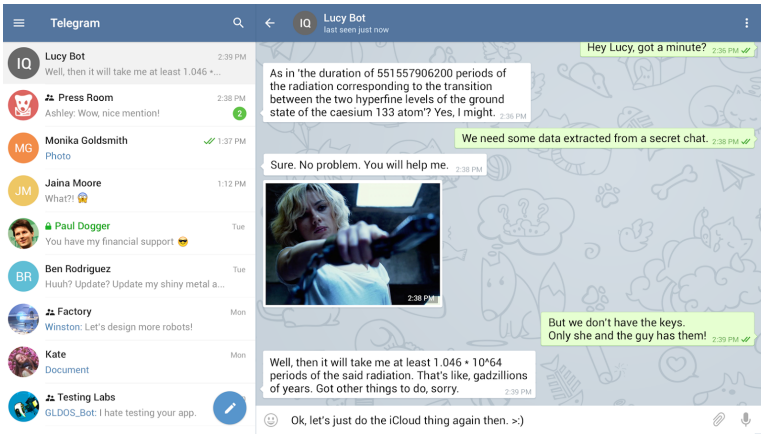
- Communication in Teams in so-called channels
- More structured than Hangouts, less messy than facebook, simpler than IRC
- Advantages
  - Many functions and hooks
  - Good connectivity with other systems (git commit-messages)
  - Even free accounts quite powerful
- Disadvantages
  - …

Mattermost



☞ mattermost.org
Self-hosted Slack Clone, bei gitlab “dabei”

Telegram



☞ telegram.org

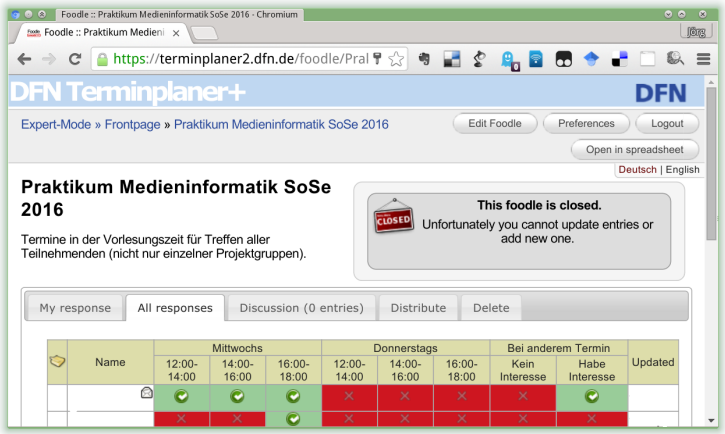
Messenger, End-to-End-Encryption bei privaten chats, Gruppen

Signal

☞ whispersystems.org

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

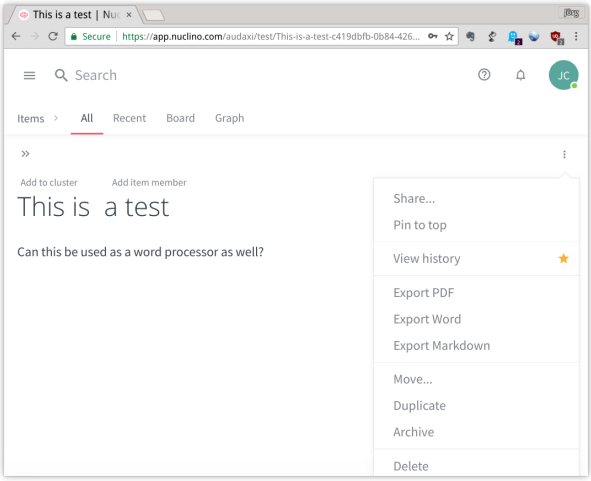
Foodle



☞ terminplaner2.dfn.de

Terminplaner, DFN

Nuclino



☞ nuclino.com

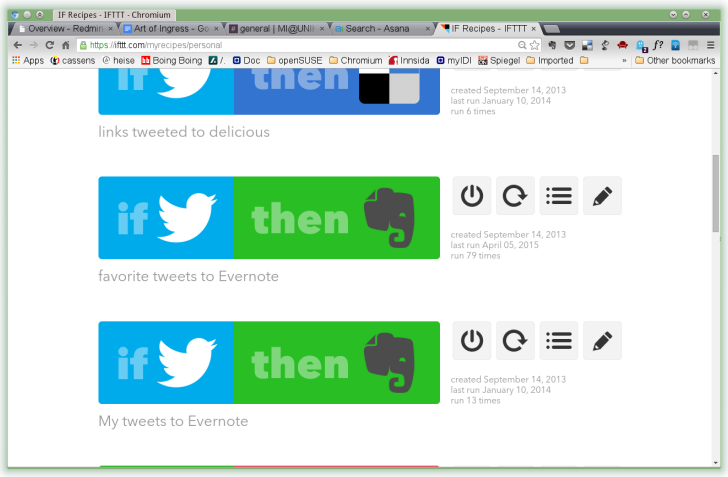
"Easy knowledge base for teams", includes collaborative word processor

4.4 Automation

Hooks

- 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

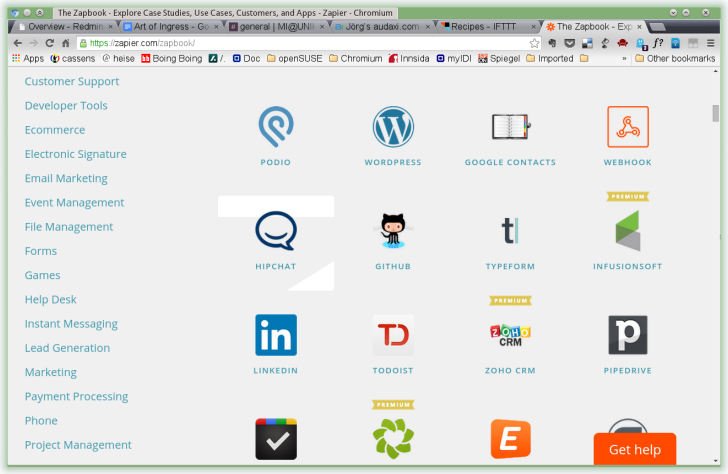
if this then that



☞ ifttt.com

If this then that, commercial, freemium

zapier



☞ zapier.com

Similar to ifttt, commercial, freemium

ifttt, zapier

- Both services make it possible to connect different data sources and data sinks from different services
- iftttt 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

4.5 Suggestions

Suggestions

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

References

git: Info & Tools

- 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

git: Hosting

- 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
  - ☞ LaTeX & ☞ git
  - ☞ 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, Automation

- 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