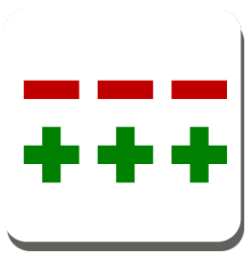
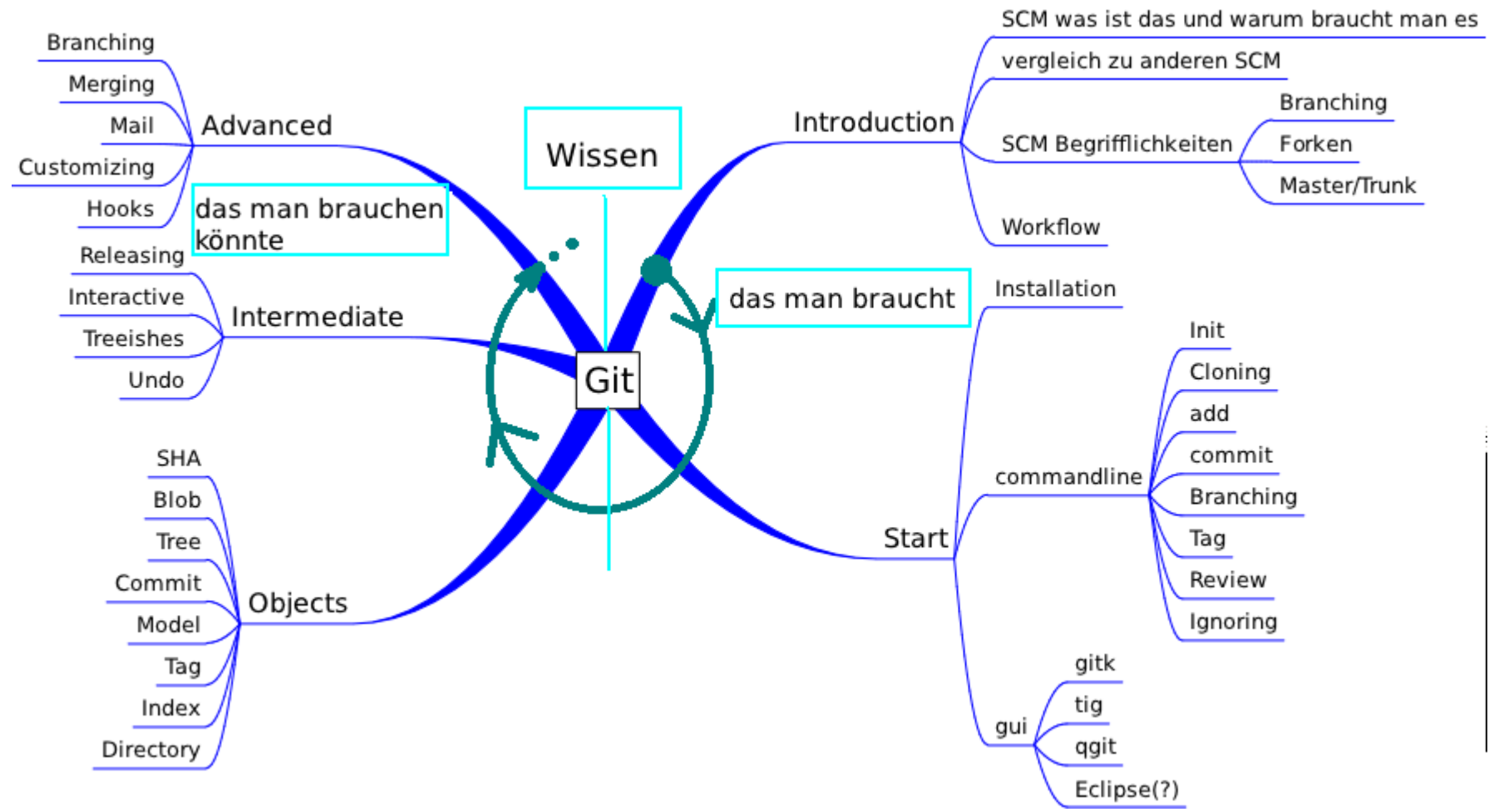
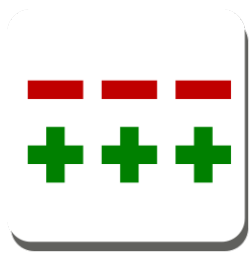


Git



# Git nutzen !

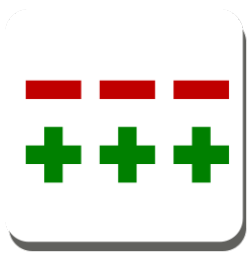




# Introduction

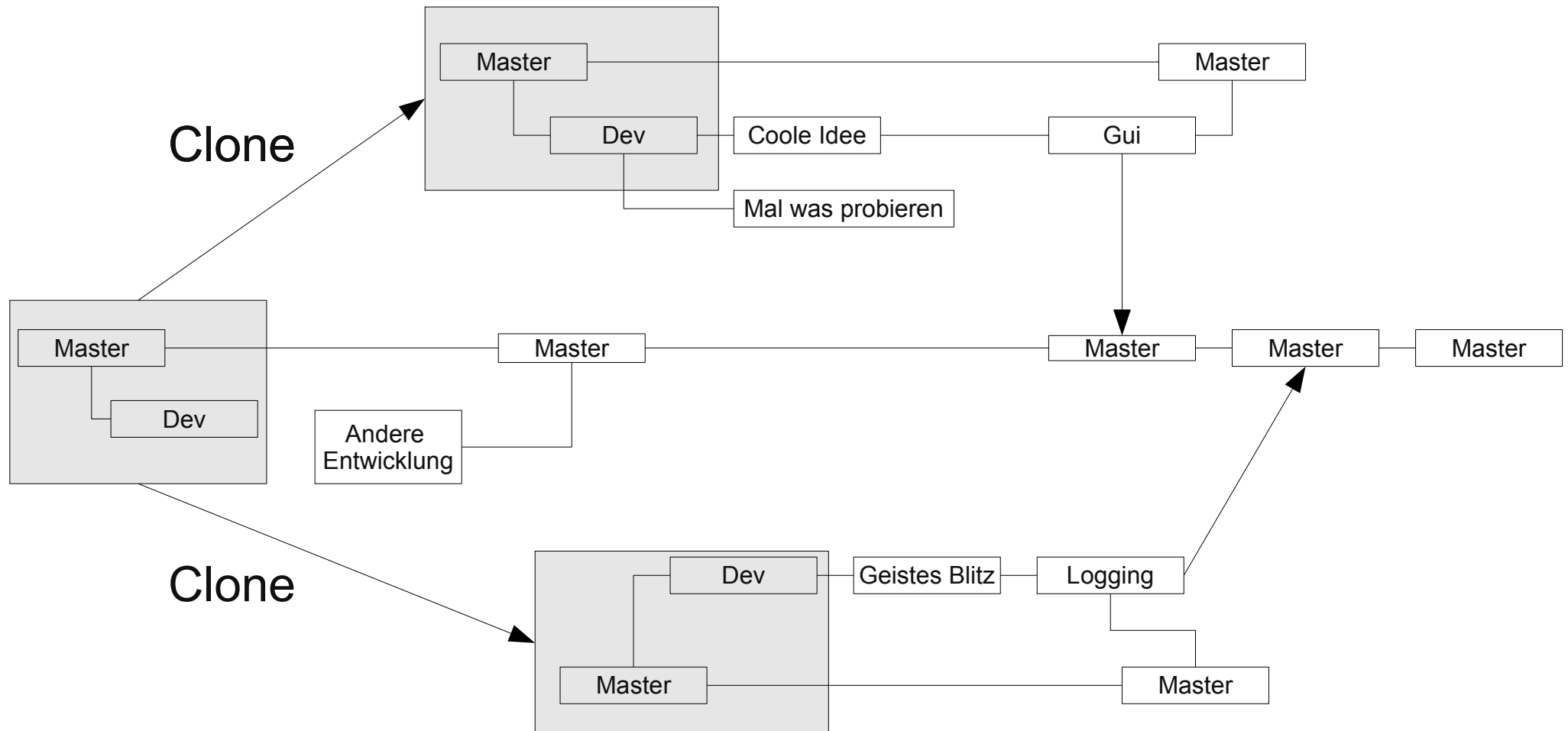
SCM was ist das und warum braucht man es braucht

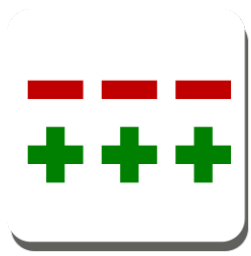
- Clean Code Basics
  - Keine Wiederholungen
  - Einfach halten
  - ...
- Source Control System (SCM)
  - Kein Angst vor Änderungen, Probieren, Tests, Lernversuchen
  - Dezentrale Weiterentwicklung (Git)
  - Verschiedene Entwicklungs Pfade ohne Verlust des Bezugs zur Basis (Git)



# Introduction

## Workflow

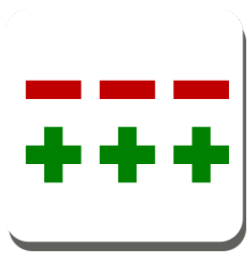




# Introduction

## Vergleich zu andere SCM

- Mercurial / Basar (close)
- Subversion/CVS (nicht flexibel genug)
- Non Free (was nichts kostet kann nichts taugen...)
  - TFS (Collaboration, Grafik, Buildsystem usw. ist dabei)
  - Clearcase (wie TFS aber noch umständlicher)



# Start

## Installation

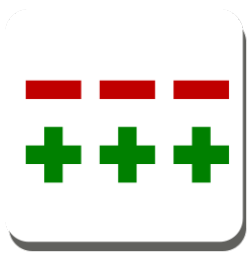
- Windows
- MAC OS
- Linux

use msysGit

```
$ sudo port install git-core /git osx installer
```

```
apt-get install git-core /yum install git-core
```

- Git config
  - \$ git config --global user.name "Scott Chacon"
  - \$ git config --global user.email "schacon@gmail.com"



# Start

## Commandline

- Init
- Cloning
- Add
- Commit
- Branching
  
- Tag
- (Reviews)
  
- Ignoring

`git init .`

`git clone %URL% / git clone path`

`git add file1 file2 file3 / git add`

`git commit -a`

`git branch (aktuelle anzeigen) / git branch %BRANCHNAME% (anlegen)`

`git pull %BRANCHNAME% / git push %BRANCHNAME%`

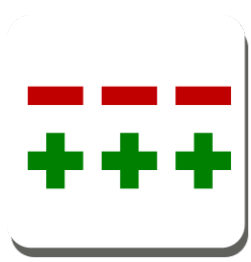
`git checkout %BRANCHNAME% (zu Branch ... wechseln)`

`git tag MEINKOMMENTAR %commitID%`

`git diff (zur letzten Version)/ git log (versionen zueinander)`

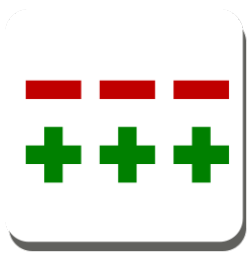
`git status (Änderungen im ganzen Baum)`

`gitignore File anlegen`



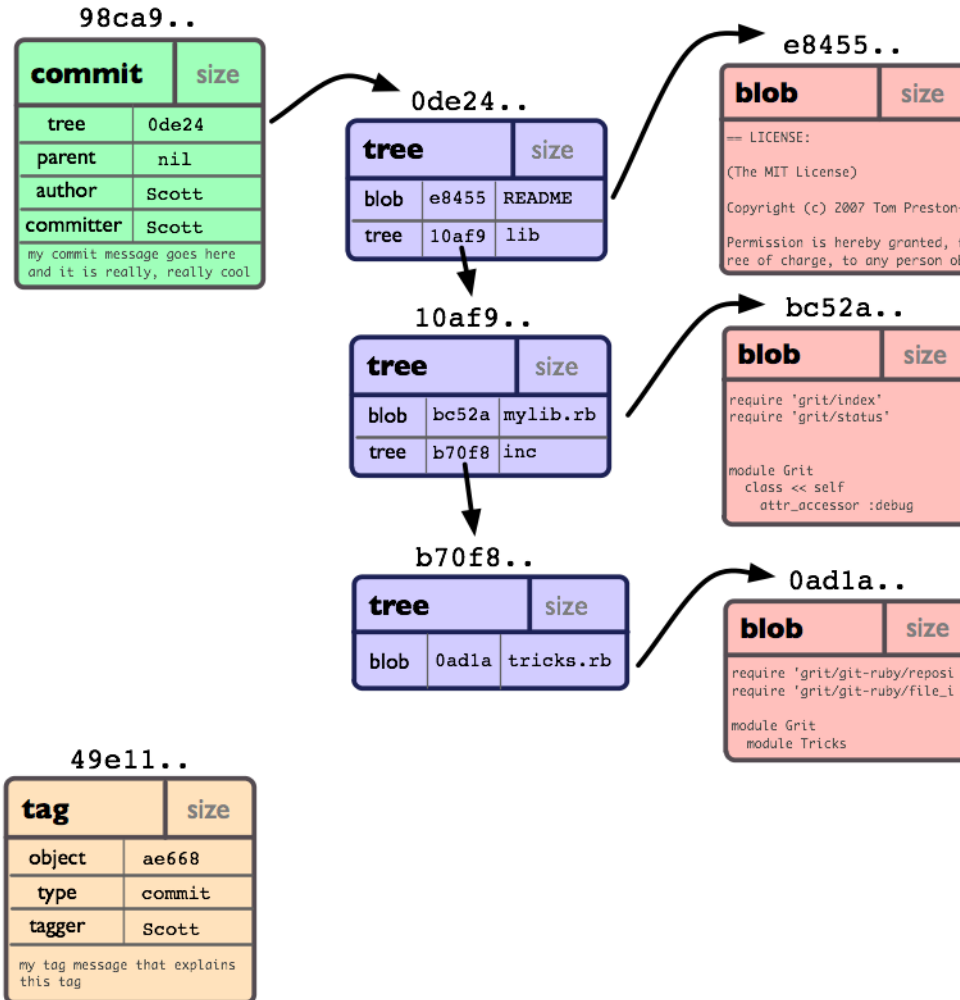
# Start

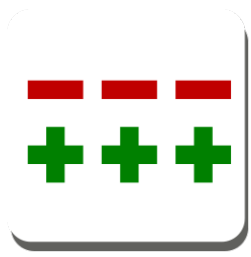
- gui
  - gitk
  - tig
  - qgit
  - Eclipse(?)



# Objects

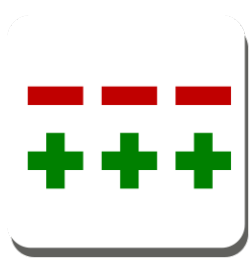
- SHA
- Blob
- Tree
- Commit
- Model
- Tag
- Index
- Directory





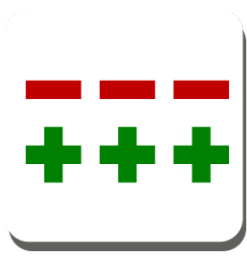
# Intermediate

- Releasing
- Interactive
- Treeishes
- Undo



# Advanced

- Branching
- Merging
- Mail
- Customizing
- Hooks



# Links im Netz

- Clean Code Initiative <http://www.clean-code-developer.de>
- Git Community Book <http://book.git-scm.com/index.html>
- Creative Commons Lizenz <http://creativecommons.org/licenses/by/3.0/de/>
- Git Source Control System <http://git-scm.com/>
- Testsystem mit Git <http://www.interaktionsdesigner.de/2009/08/31/ein-sauberer-testsystem-mit-git/>
- Git Host (auch Hackerspace) <http://github.com/>
- Ausführliche deutsche Anleitung <http://gitmagic.lordofbikes.de/>
- Git Hilfe <http://help.github.com/>
- Kurze Anleitungen [http://wiki.sourcemage.org/Git\\_Guide](http://wiki.sourcemage.org/Git_Guide)
- Wikipedia Artikel <http://de.wikipedia.org/wiki/Git>
- 10 Tutorials <http://sixrevisions.com/resources/git-tutorials-beginners/>
- Survival Guide <http://flanders.co.nz/2009/03/21/gitgithub-survival-guide/>
- Hackerspace Seite zu Git use [http://hackerspace-ffm.de/wiki/index.php?title=Git\\_Benutzen](http://hackerspace-ffm.de/wiki/index.php?title=Git_Benutzen)