

# **git im Kontext industrieller Softwareentwicklung**

Arlo O'Keeffe  
mailto:arlo@gmail.com  
2018-10-25

THIS IS GIT. IT TRACKS COLLABORATIVE WORK  
ON PROJECTS THROUGH A BEAUTIFUL  
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL  
COMMANDS AND TYPE THEM TO SYNC UP.  
IF YOU GET ERRORS, SAVE YOUR WORK  
ELSEWHERE, DELETE THE PROJECT,  
AND DOWNLOAD A FRESH COPY.



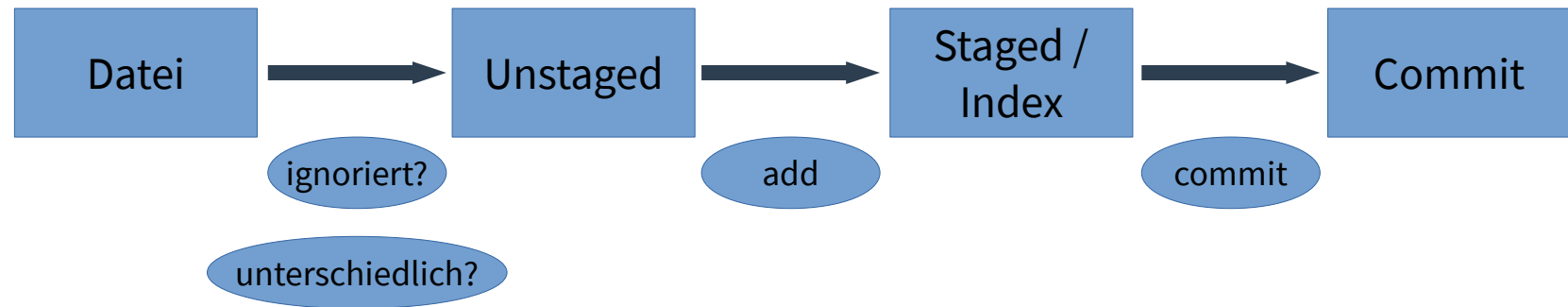
# Ziel

- Know your tools
- Alle Fragen kennen und beantworten können

# Begriffe

- Was ist eine untracked Datei?
- Was ist eine Änderung?
- Was ist die Staging?
- Was ist der Index?
- Was ist ein Commit?
- Was ist ein Hash?

# Begriffe



# Begriffe

- Was ist ein Branch?
- Was ist ein Tag?
- Was ist der HEAD?
- Was ist ein detached HEAD?
- Was ist HEAD^?
- Was ist HEAD^^?
- Was ist HEAD~3?

# Begriffe

- Was ist ein Checkout?
- Was ist ein Clone?
- Was ist ein Repository?
- Was ist ein bare Repository?
- Was ist ein Remote?

# Begriffe

- Was ist ein Stash?
- Was ist eine Hook?
- Was ist ein Submodule?
- Was ist ein Pull Request?



# Aktionen

- Wie erstelle ich ein neues Repository?
- Wie sehe ich lokale Änderungen?
- Wie stage ich eine Änderung?
- Wie stage ich nur einen Teil meiner Änderungen?
- Wie committe ich meine Änderungen?
- Wie stage und committe ich eine Änderung?

# Aktionen

- Wie ignoriere ich bestimmte Dateien?
- Wie ignoriere ich einen ganzen Ordner?
- Wie de-ignoriere ich bestimmte Dateien in einem ignorierten Ordner?
- Wie füge ich trotz .gitignore eine Datei hinzu?

# Aktionen

- Wie erstelle ich einen Branch?
- Wie lösche ich einen Branch?
- Wie wechsel ich den aktuellen Branch?
- Wie sehe ich den Unterschied zwischen zwei Branches?
- Wie merge ich zwei Branches?
- Wie löse ich einen Merge-Konflikt?
- Wie erstelle ich einen Tag?

# Aktionen

<<<<<<<

Mine. Changes made on the branch that is being merged into. In most cases, this is the branch that I have currently checked out (i.e. HEAD).

|||||

The common ancestor version.

=====

Theirs. Changes made on the branch that is being merged in. This is often a feature/topic branch.

>>>>>>>

# Aktionen

- Wie mache ich eine Änderung rückgängig?
- Wie ändere ich eine Commit Message?

# Aktionen

- Wie hole ich mir ein externes Repository?
- Wie schiebe ich meine Änderungen in ein externes Repository?
- Wie aktualisiere ich den Zustand vom externen Repository?
- Wie hole ich mir die neusten externen Änderungen in mein Projekt?

# Aktionen

- Wie kann ich Änderungen zwischenspeichern?
- Wie kann ich Änderungen zwischenspeichern und den Index behalten?
- Wie kann ich einen einzelnen Commit eines anderen Branches übernehmen?
- Wie entferne ich alle Änderungen und unnötige Dateien (inkl. ignorierten)?

# Aktionen

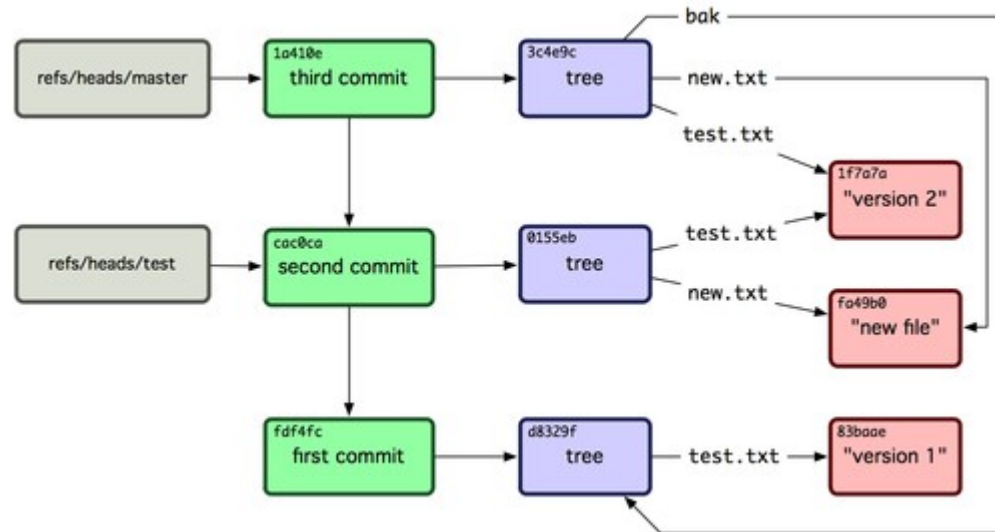
- Wie finde ich raus wer das commitet hat?
- Wie finde ich raus wann der Fehler eingeführt wurde?
- Wie füge ich unter Windows ein Executable Flag hinzu?
- Wie entferne ich nicht erreichbare Commits?
- Wie finde ich nicht erreichbare Commits?



# Prinzipien

- Was ist ein Graph?
- Was ist ein gerichteter Graph?
- Was ist ein azyklischer Graph?
- Was ist Erreichbarkeit?
- Was ist ein Baum?
- Was ist ein Object?
- Was ist eine Referenz?

# Prinzipien



# Prinzipien

- Was ist ein Event-Sourced System?
- Was bedeutet es, dass git verteilt ist?
- Kann ich ein Git Repository zippen?
- Was ist porcelain?
- Was ist plumbing?

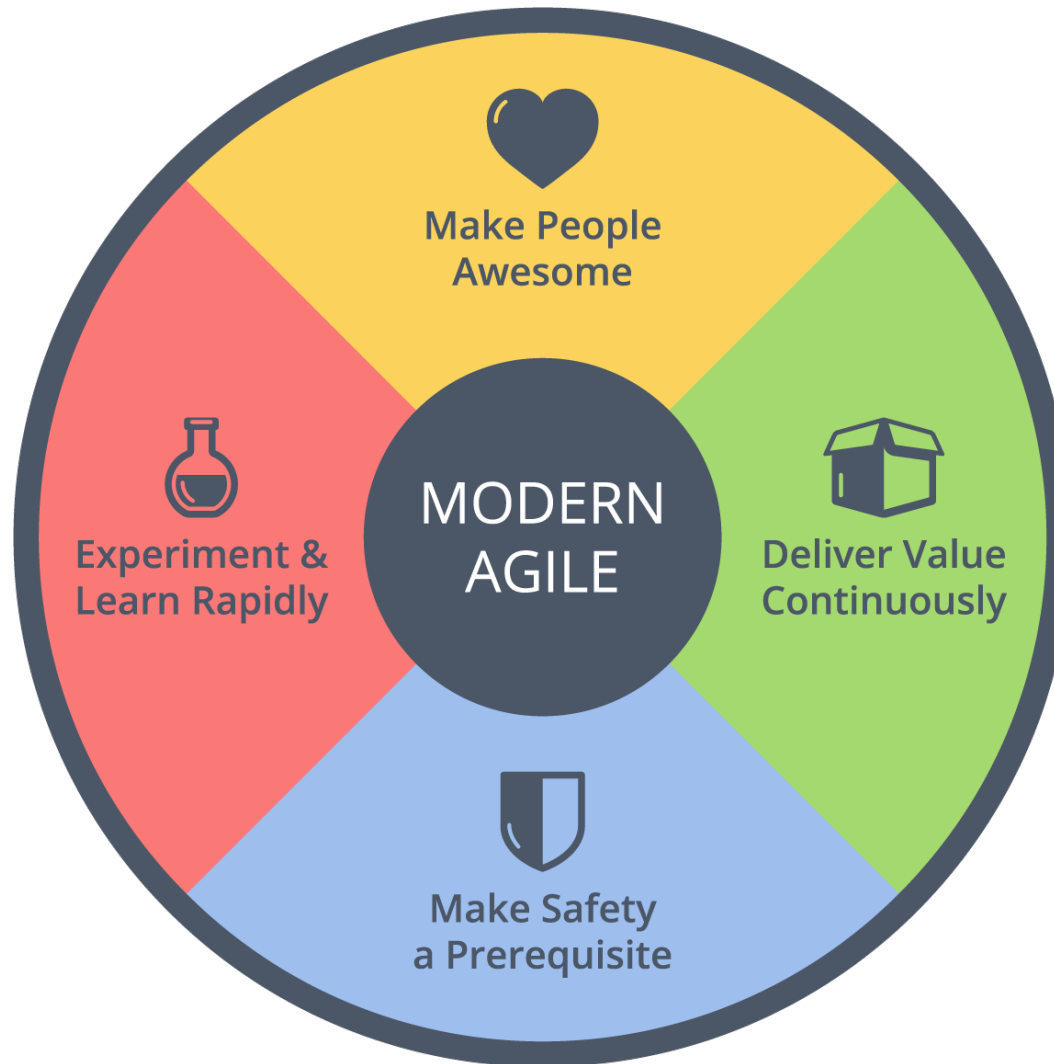
# Philosophie

- Warum will man eine Versionskontrolle?
- Ist git die beste Versionskontrolle?
- Ist git benutzerfreundlich?
- Ist git besser als svn?
- Was ist Social Coding?

# Kollaboration

- Wie arbeitet man mit mehreren Leuten zusammen?

# Kollaboration



# Kollaboration

- Welcher Workflow wird eingesetzt?
- Was sind sonstige Konventionen?
- Wie sieht der Prozess zwischen finalem Commit und Release zum End-User aus (aka Deploy)?
- Gibt es Reviews? Wer reviewed wen?
- Wo ist das Haupt-Repository gehostet?

# Repository Management

- Was will man alles in einem Repository haben?
- Was gehört in eine .gitignore?
- Was gehört in eine gitattributes?
- Wie Sorge ich dafür, dass in jedem Betriebssystem die richtigen Line Endings sind?



# Branch Management

- Wie werden Branches benannt?
- Wofür werden Branches erstellt?
- Wie sieht der Lebenszyklus eines Branches aus?
- Wer macht was? Wer arbeitet wo?
- Sind Notfall-Patches notwendig?
- Müssen alte Releases supported werden?
- Was ist ein protected Branch?

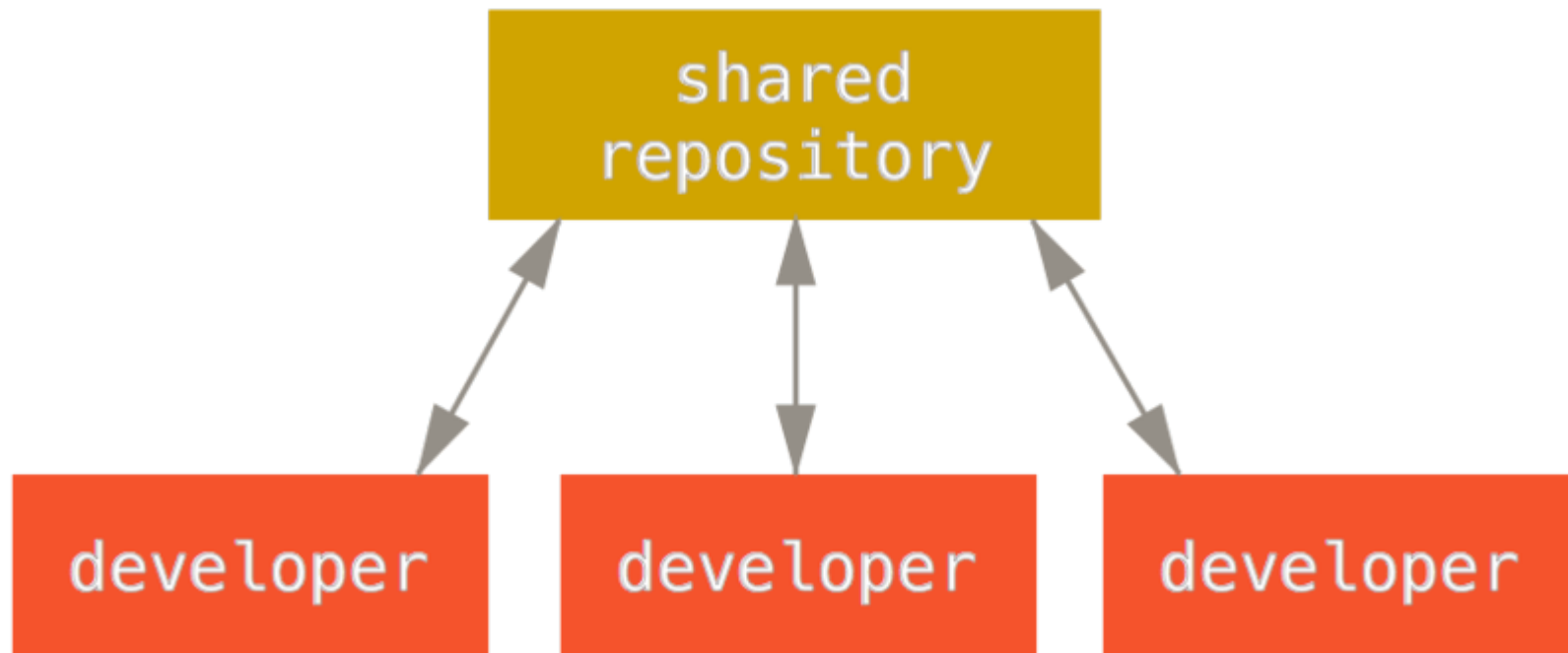
# Tag Management

- Wie werden Tags benannt?
- Wofür werden Tags erstellt?
- Wer erstellt Tags?

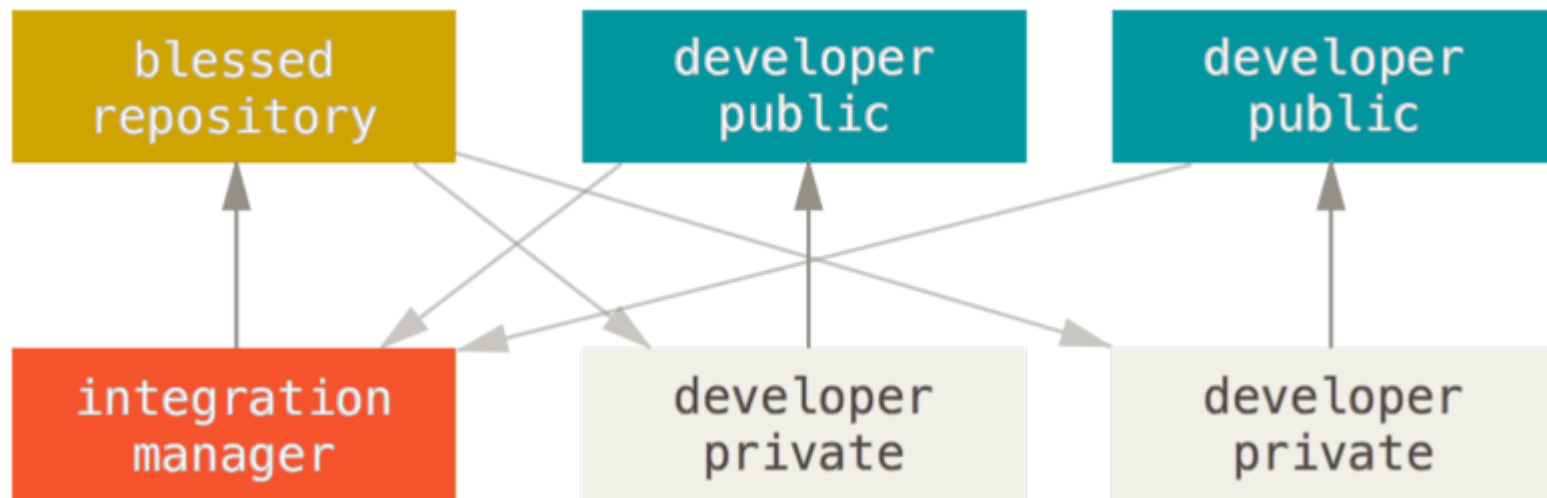
# Log / History Management

- Wie sehen Commit Messages aus?
- Interessiert mich das „Sausage Making“?
- Wie oft wird committed?
- Wird ge-squashed?
- Werden Branches gerebased oder wird Baseline gemerged?

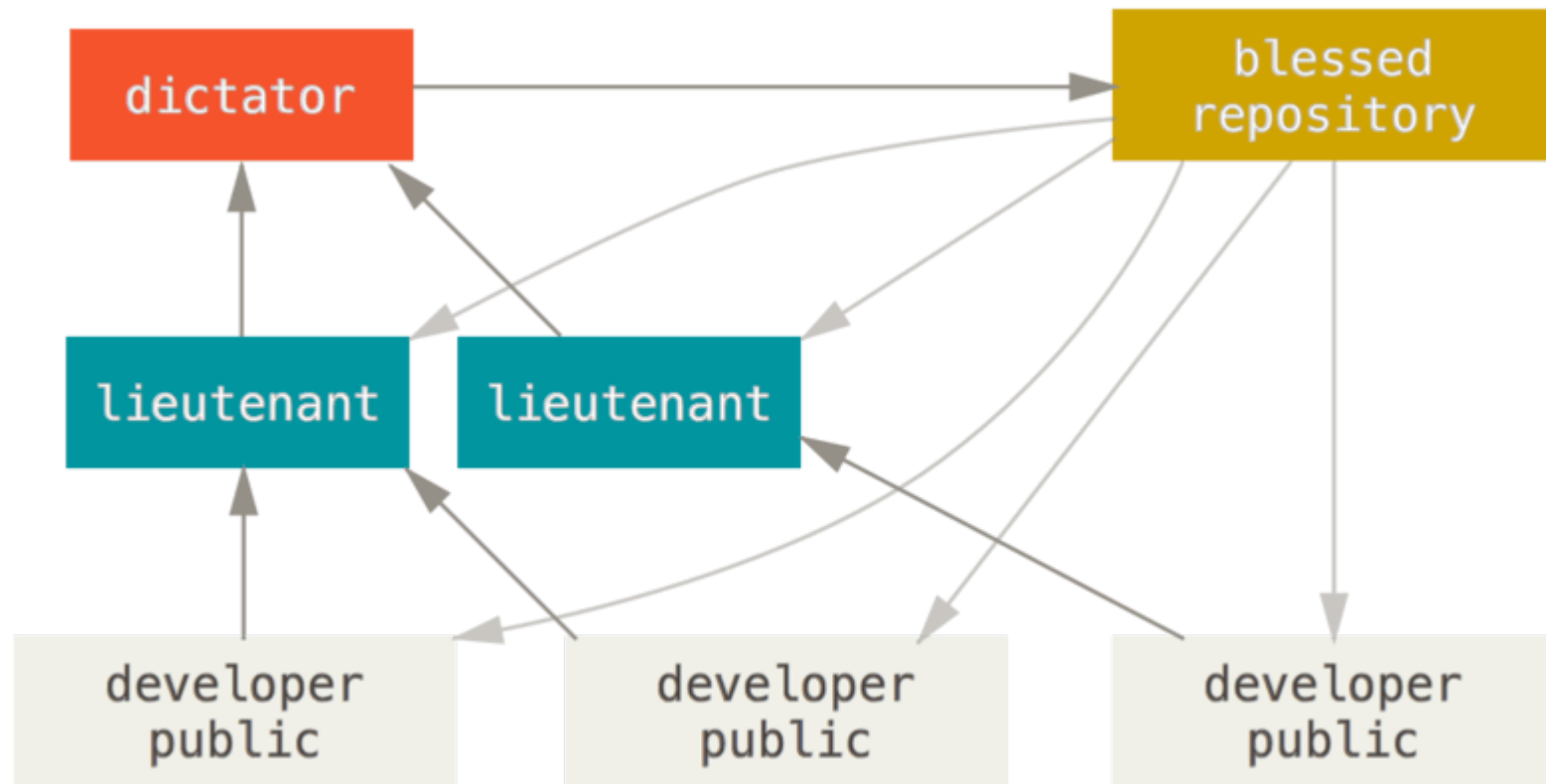
# Kollaboration – Zentralisierter Workflow



# Kollaboration – Integration Manager

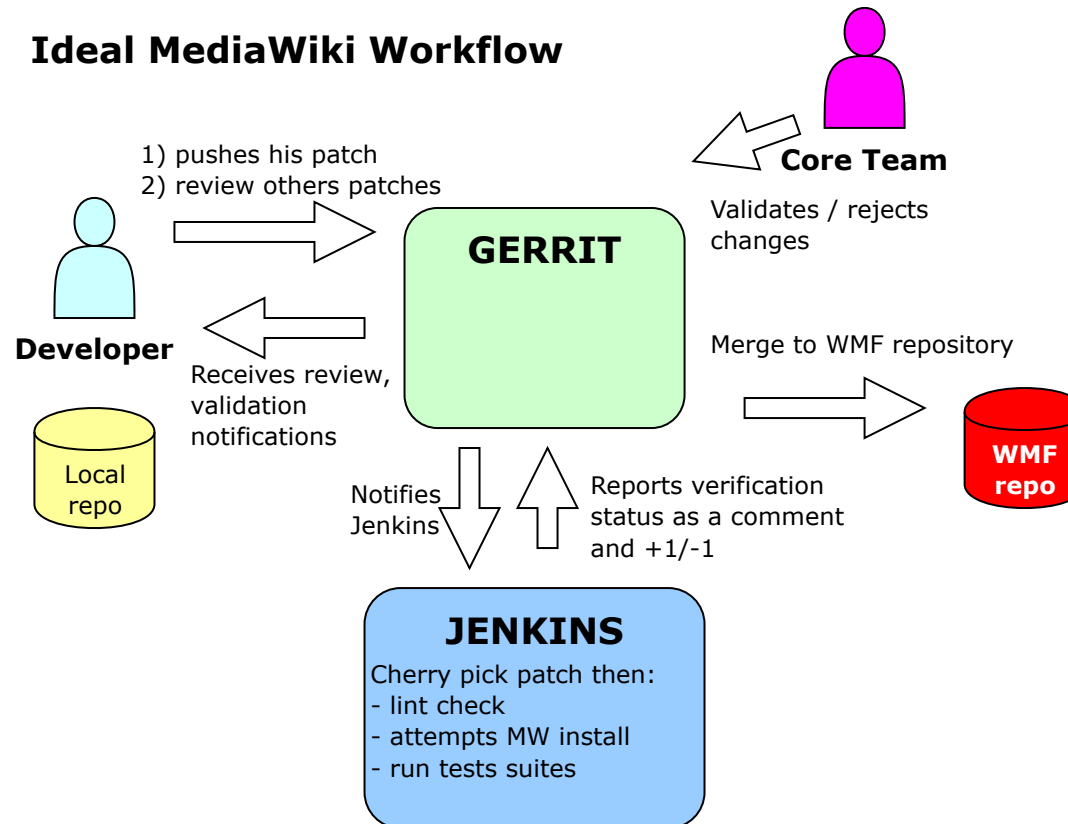


# Kollaboration - Benevolent Dictator

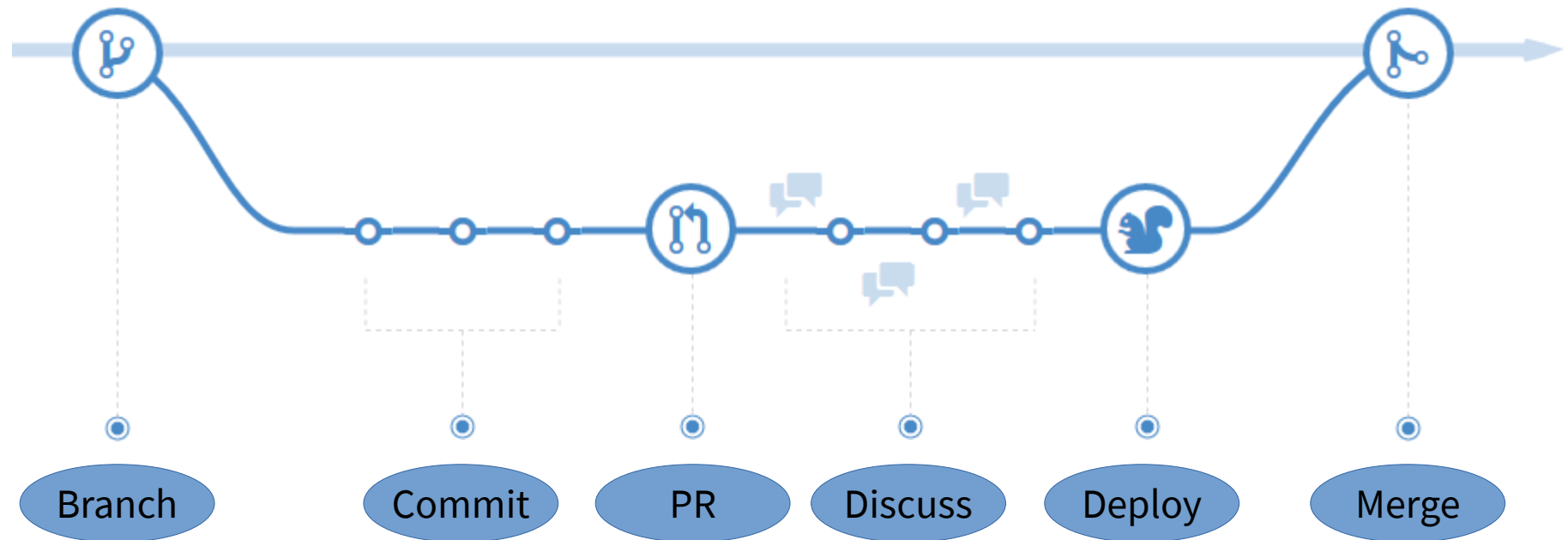


# Kollaboration - Gerrit

## Ideal MediaWiki Workflow

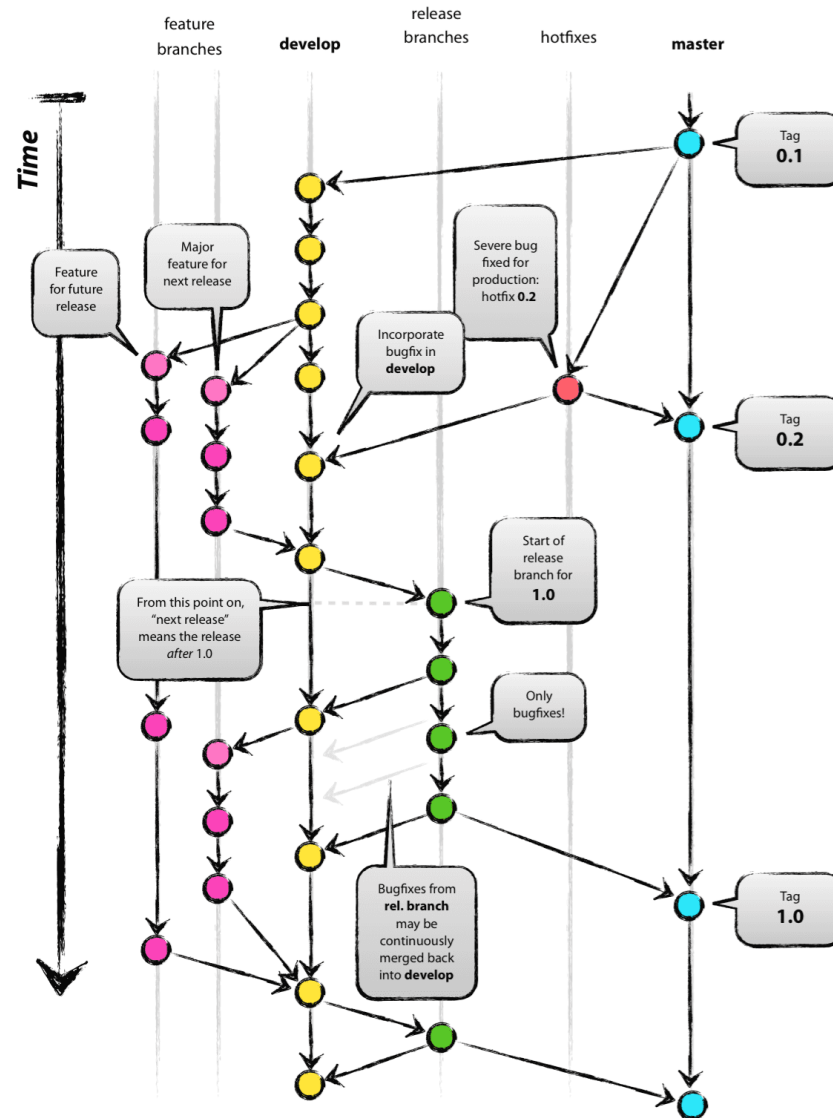


# Kollaboration - GitHub Flow





# Kollaboration - Git Flow



# Alltag

- Backups
- Unterschiedliche Clients (GUI vs. Nicht-GUI)
- Was ist die .gitconfig?
- Was ist autostash?
- Was ist eine globale .gitignore?
- Was sind Aliase?

# Einstellungen

- `user.name=Arlo O'Keeffe`
- `user.email=mailto:arlo@gmail.com`
- `core.autocrlf=true`
- `core.excludesfile=~/.gitignore_global`
- `color.ui=true`
- `push.default=simple`
- `rebase.autostash=true`
- `pull.ff=only`
- `pull.rebase=true`
- `alias.adog=log --all --decorate --oneline --graph`

# Automatisierung und Integration

- Skripting via Bash / Batch
- JGIT
- Atlassian JIRA + BitBucket
- Fix / Close Commits
- Continuous Deployment
  - .git Ordner nicht veröffentlichen

**mailto:carlo@gmail.com**