Git Workshop
Prinzipien und Funktion

GECKO | a WIIT Company

GECKO a WIIT Company

Git Workshop

Wer bin ich?

STEFFEN SACHSE

Softwareentwickler mit Fokus auf Webanwendungen, spezialisiert auf Frontends und Progressive Web Apps

+49 381 45488 749
steffen.sachse@gecko.de

Gliederung

1. Einführung in Versionskontrolle
2. Installation und Konfiguration
3. Grundlegende Git-Befehle
4. Arbeiten mit Branches
5. Remote Repositories
6. Konfliktlösung
7. Workflows

GECKO | a WIIT Company

Gliederung

1. **Einführung in Versionskontrolle**
2. Installation und Konfiguration
3. Grundlegende Git-Befehle
4. Arbeiten mit Branches
5. Remote Repositories
6. Konfliktlösung
7. Workflows

GECKO | a WIIT Company

GECKO a WIIT Company

Git Workshop

Was ist Versionskontrolle?

- **Nachvollziehbarkeit**
  - **Welche Erweiterungen sind in das Projekt in welcher Art eingeflossen**
- **Wiederherstellbarkeit**
- **Kontrollierbarkeit**
  - **Wer hat wann was beigetragen**
- **Kollaboration**
  - **Arbeiten in Teams**
  - **Unabhängig**
  - **Verwaltbar**

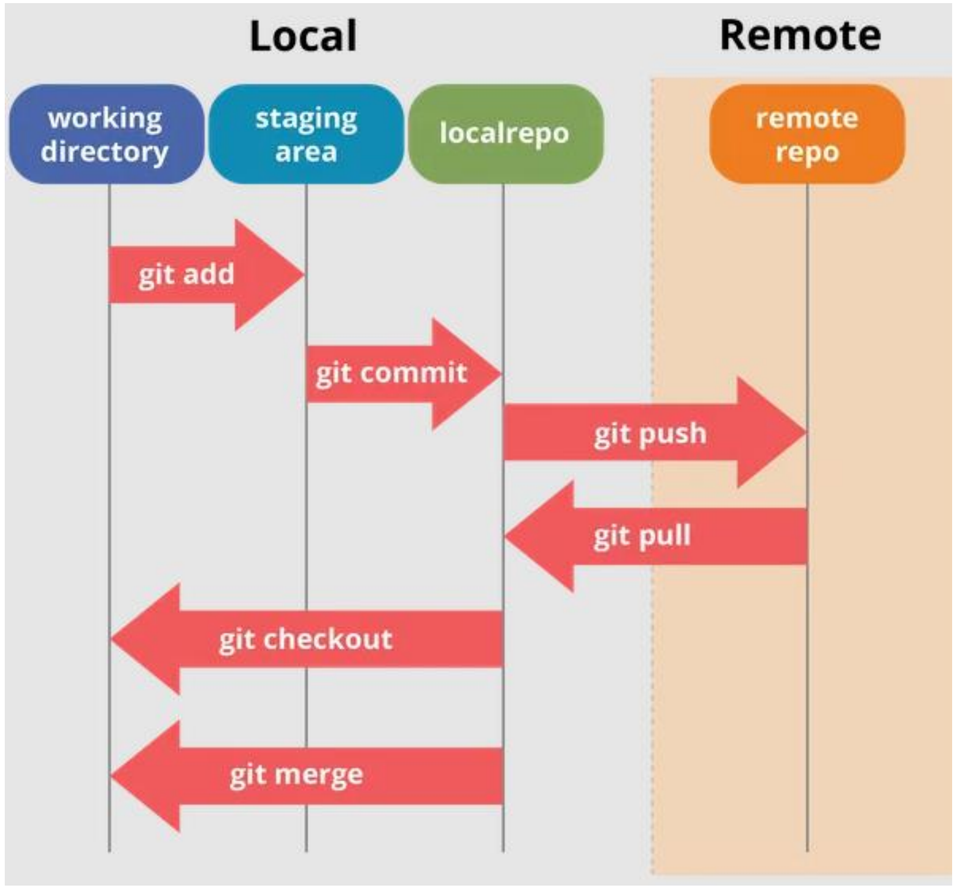
Git Workshop

Was ist Git? #2

- **Arbeitsverzeichnis (Working Directory):**
  - Das Arbeitsverzeichnis ist der Ort, an dem aktiv am Projekt gearbeitet wird. Hier befinden sich Dateien und Ordner.
- **Staging-Bereich (Index):**
  - Der Staging-Bereich ist eine Zwischenstufe zwischen Arbeitsverzeichnis und dem Repository. Man verwendest ihn, um bestimmte Änderungen auszuwählen und für den nächsten Commit vorzumerken.
- **Repository (Local Repository):**
  - Das Repository ist der Ort, an dem die vollständige Versionsgeschichte des Projekts gespeichert wird. Es enthält alle Commits, Branches, Tags und andere Metadaten, die zur Verwaltung deines Codes benötigt werden. Das lokale Repository wird auf dem eigenen Computer gespeichert
- **Remote Repository**
  - Zum kollaborativen Arbeiten

Git Workshop

Was ist Git? #3



Quelle

Gliederung

1. Einführung in Versionskontrolle
2. **Installation und Konfiguration**
3. Grundlegende Git-Befehle
4. Arbeiten mit Branches
5. Remote Repositories
6. Konfliktlösung
7. Workflows

GECKO | a WIIT Company

Git Workshop

Installation und Konfiguration

Git-for-Windows

https://git-scm.com/download/win

Installation unter Debian

```
$ apt-get install git
```

Benutzername und E-mail

```
$ git config --global user.name „John Doe“
$ git config --global user.email johndoe@example.com
```

Git Workshop

Installation und Konfiguration #2

Konfigurationsdatei im Home-Verzeichnis (~/.gitconfig) – Exemplarischer Auszug

```
[alias]
    last = log -1 HEAD
    lg = log --date-order --graph --format='%C(green)%h%Creset %C(yellow)%an%Creset %C(blue bold)%ar%Creset %C(red bold)%d%Creset%s'
    lga = log --all --date-order --graph --format='%C(green)%h%Creset %C(yellow)%an%Creset %C(blue bold)%ar%Creset %C(red bold)%d%Creset%s'
    lgad = log --all --date-order --graph --format='%C(green)%h%Creset %C(yellow)%an%Creset %C(blue bold)%cd%Creset %C(red bold)%d%Creset%s'
    decorate = log --all --date-order --graph --format='%C(green)%h%Creset %C(yellow)%an%Creset %C(blue bold)(%ci)%ar%Creset %C(red bold)%d%Creset%s' --simplify-by-decoration
    lgd = log --date-order --graph --format='%C(green)%h%Creset %C(yellow)%an%Creset %C(blue bold)%ad%Creset %C(red bold)%d%Creset%s'
    st = status
    hidden = !git ls-files -v | grep ^[a-z]
    addnw = !sh -c 'git diff -U0 -w --no-color "$@" | git apply --cached --ignore-whitespace --unidiff-zero -'
[diff]
    tool = p4merge
[difftool "p4merge"]
    path = C:\\Program Files\\Perforce\\p4merge.exe
[merge]
    tool = p4merge
[mergetool "p4merge"]
    path = C:\\Program Files\\Perforce\\p4merge.exe
```

Kann auch pro Repository verwendet werden!

Gliederung

1. Einführung in Versionskontrolle
2. Installation und Konfiguration
3. **Grundlegende Git-Befehle**
4. Arbeiten mit Branches
5. Remote Repositories
6. Konfliktlösung
7. Workflows

GECKO | a WIIT Company

Git Workshop

Grundlegende Befehle

- **git init**
  - Anlegen eines Git-Repositories
  - (--bare für ein reines Remote-Repository, mehr dazu später)
- **git clone**
  - Legt eine lokale Kopie eines Repositories an
- **git status**
  - Zeigt den aktuellen Zustand des Repositories an
  - Immer guter Ausgangspunkt bei merge-Vorgängen oder anderen „Problemen“
- **git add (-p)**
  - Hinzufügen von Änderungen zur „Staging Area“
  - Ganze Dateien oder nur Teile von geänderten Inhalten

GECKO a WIIT Company

Git Workshop

Grundlegende Befehle

- **git reset (-p)**
  - Zurücksetzen des Arbeitsbereiches
  - Ganze Dateien oder nur Teile von geänderten Inhalten
- **git commit**
  - Erstellen von einem neuen Commit lokal
  - Ganze Dateien oder nur Teile von geänderten Inhalten
- **git show**
  - Anzeigen des letzten Commits
- **git log**
  - Anzeigen der Versionshistorie des Repositories

Gliederung

1. Einführung in Versionskontrolle
2. Installation und Konfiguration
3. Grundlegende Git-Befehle
4. **Arbeiten mit Branches**
5. Remote Repositories
6. Konfliktlösung
7. Workflows

GECKO | a WIIT Company

Git Workshop

Arbeiten mit Branches

- **git checkout –b / git switch**
  - Anlegen eines Git-Branches
  - Standardmäßig abgezweigt vom aktuellen Versionierungspunkt/Branch
- **git branch –d/-D**
  - Löschen von Branches
- **git merge**
  - Zusammenführen von Änderungen verschiedener Branches
- **git rebase**
  - Umschreiben der Commit-Historie eines Branches
    - Achtung: Vorsicht bei zusammenarbeit!

Gliederung

1. Einführung in Versionskontrolle
2. Installation und Konfiguration
3. Grundlegende Git-Befehle
4. Arbeiten mit Branches
5. **Remote Repositories**
6. Konfliktlösung
7. Workflows

GECKO | a WIIT Company

Git Workshop

Arbeiten mit Remote Repositories

- **git remote**
  - Verwalten der Konfiguration eines Servers
  - Hinzufügen weiterer Endpunkte (dezentral)
- **git fetch**
  - Herunterladen von Änderungen
- **git pull**
  - Anwenden der Änderungen auf den aktuellen Branch

Gliederung

1. Einführung in Versionskontrolle
2. Installation und Konfiguration
3. Grundlegende Git-Befehle
4. Arbeiten mit Branches
5. Remote Repositories
6. **Konfliktlösung**
7. Workflows

GECKO | a WIIT Company

GECKO a WIIT Company

Git Workshop

Konfliktlösung

Konflikt entsteht, wenn mehrere Personen oder Branches an der gleichen Stelle Änderungen vornehmen, beim Zusammenführen

- Grundlegende Operation bei Zusammenarbeit
- Gute Tools können helfen

```
<<<<<<< HEAD
document.addEventListener("DOMContentLoaded", function () {
  const button = document.getElementById("clicker-button");
  button.addEventListener("click", function () {
    alert("Das war nicht sehr komplex");
  });
=======
document.getElementById("click-alert").addEventListener("click", function () {
  alert("Hello world");
>>>>>>> feature/clicker-alert
});
```

Gliederung

1. Einführung in Versionskontrolle
2. Installation und Konfiguration
3. Grundlegende Git-Befehle
4. Arbeiten mit Branches
5. Remote Repositories
6. Konfliktlösung
7. **Workflows**

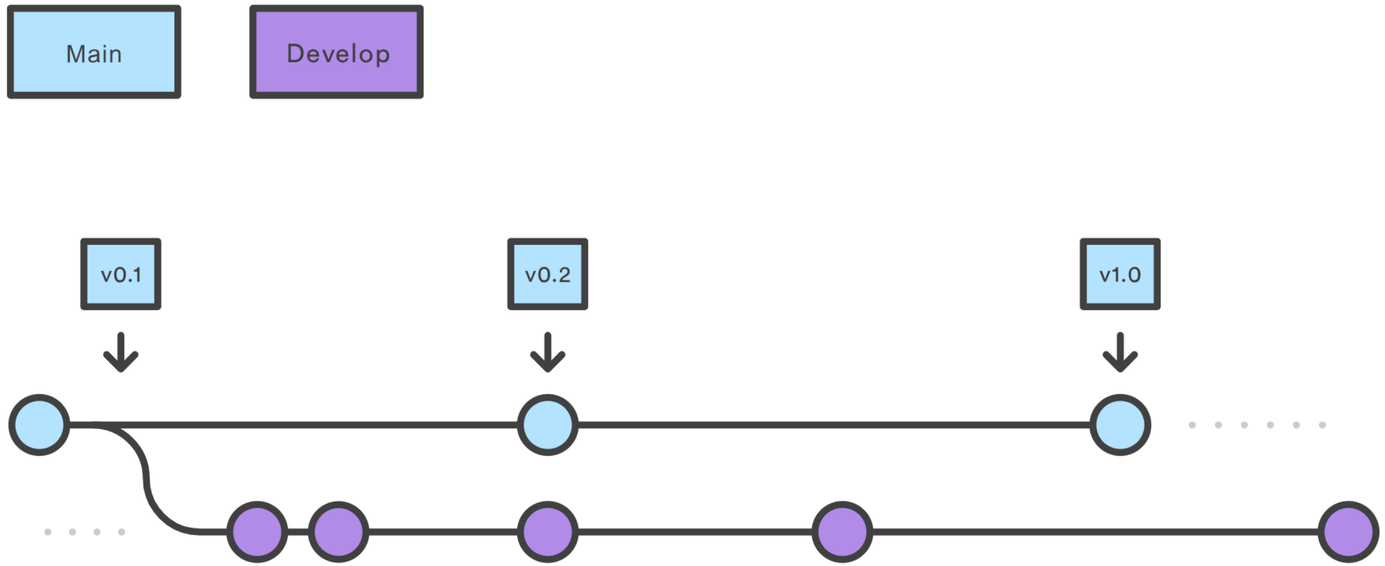
GECKO | a WIIT Company

Git Workshop

Workflows

Workflows dienen der Organisation und Einheitlichkeit

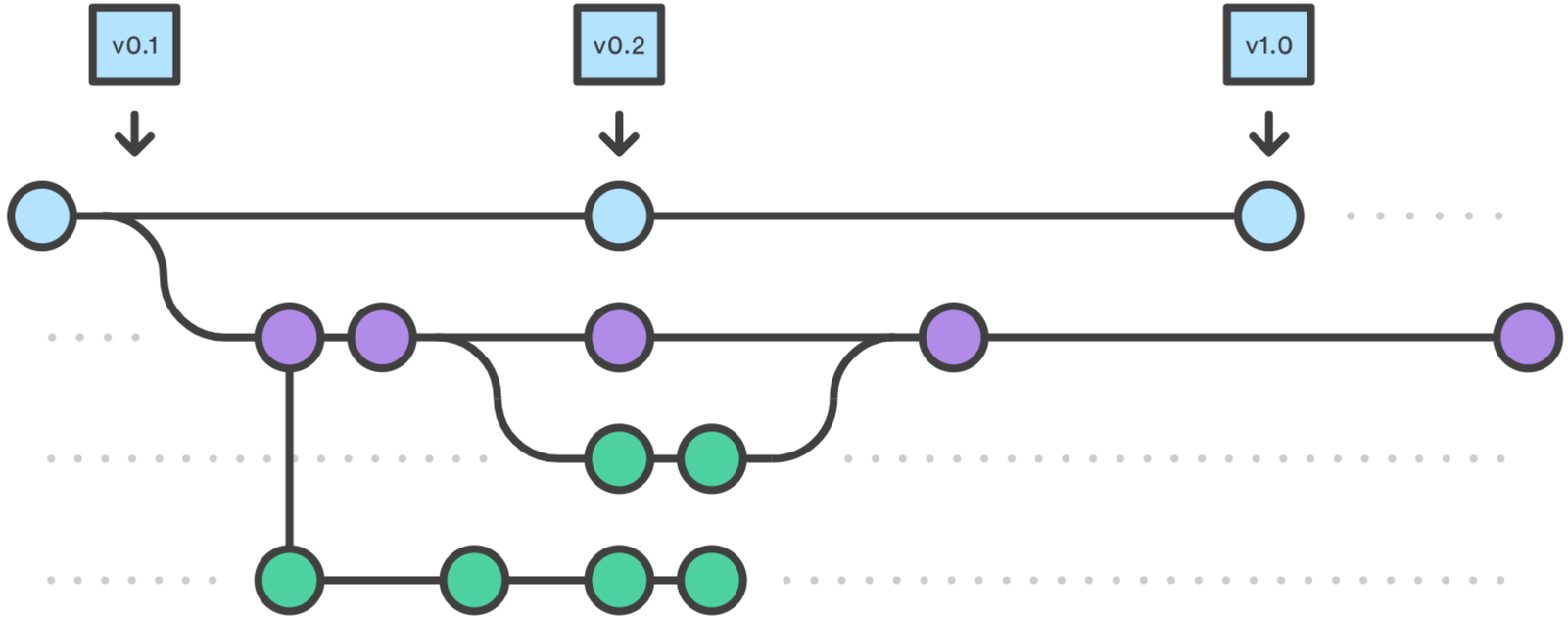
- git flow



GECKO a WIIT Company

Git Workshop

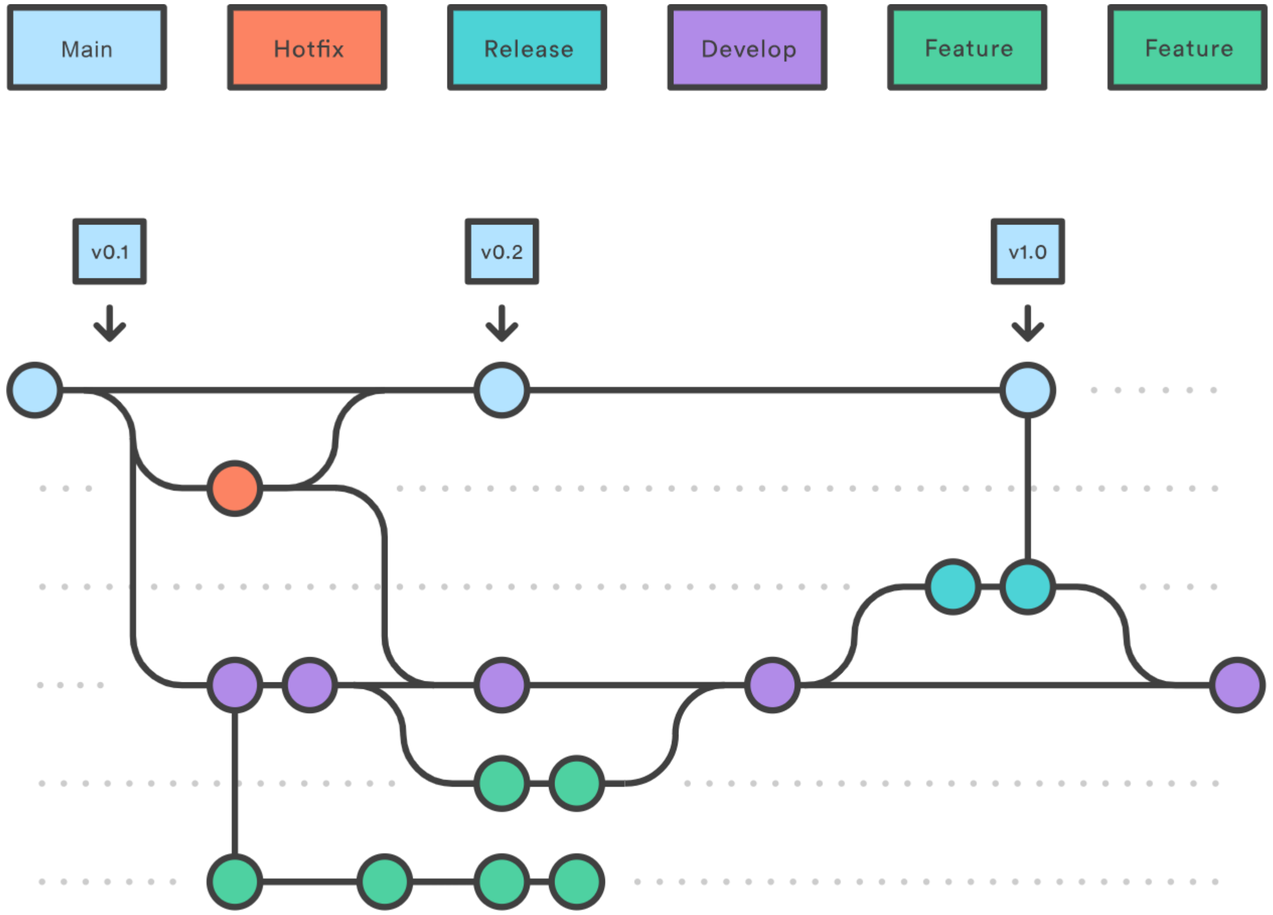
Workflows



GECKO a WIIT Company

Git Workshop

Workflows



GECKO a WIIT Company

Git Workshop

Workflows

- **viele Verschiedene Ansätze**
- **Inspiration, aber immer an eigene Anforderungen anpassen**

❖ **Atlassian**

❖ **Github**

GECKO a WIIT Company

Git Workshop

Empfehlenswerte Tools

Sourcetree: GUI für git mit umfangreichen Funktionen

Git Tower: GUI für git mit umfangreichen Funktionen, gratis für Studierende

p4merge: Visuelles Merge-Tool

GECKO a WIIT Company

Git Workshop

Weiterführende Tutorials und Links

git-it-electron:	Interaktiver Kurs in einer eigenständigen Anwendung, Open-Source
learngitbranching.js:	Webseite zum Ausprobieren von Git-Befehlen
Atlassian Tutorials:	Sammlung von Einführenden und Fortgeschrittenen Tutorials

Danke für die Aufmerksamkeit.
Let’s talk.

GECKO | a WIIT Company

Deutsche-Med-Platz 2 • 18057 Rostock
0381-454880 • info@gecko.de • gecko.de