

Seminar



Bachelor Wirtschaftsinformatik

Generalthema: Aktuelle Technologien zur
Entwicklung verteilter Java-Anwendungen

Versionierung von Java-Anwendungen mit Git

Überblick

- 1. Motivation**
2. Fachbegriffe zur Versionierung
3. Prinzip der Versionsverwaltung
4. Arten der Versionsverwaltungen
5. Branching und Merging
6. Versionierungssystem Git
7. GitHub
8. Fazit

Motivation

- Änderungen rückgängig machen
- Projekt auf vorherigen Stand zurücksetzen
- Wiederherstellung gelöschter Dateien
- Riskante Änderungen ausprobieren
- Testen experimenteller Änderungen

Faktor Teamarbeit

- Backup-Software ausreichend
- Aber: Oftmals Teamarbeit
- Folgen:
 - Mehr Anforderungen
 - Backup-Software ungeeignet
- Weitere Anforderungen an Versionierungssysteme:
 - Änderungen anderer Teammitglieder mitbekommen
 - Isolierung experimenteller Änderungen am Projekt



Überblick

1. Motivation
- 2. Fachbegriffe zur Versionierung**
3. Prinzip der Versionsverwaltung
4. Arten der Versionsverwaltungen
5. Branching und Merging
6. Versionierungssystem Git
7. GitHub
8. Fazit

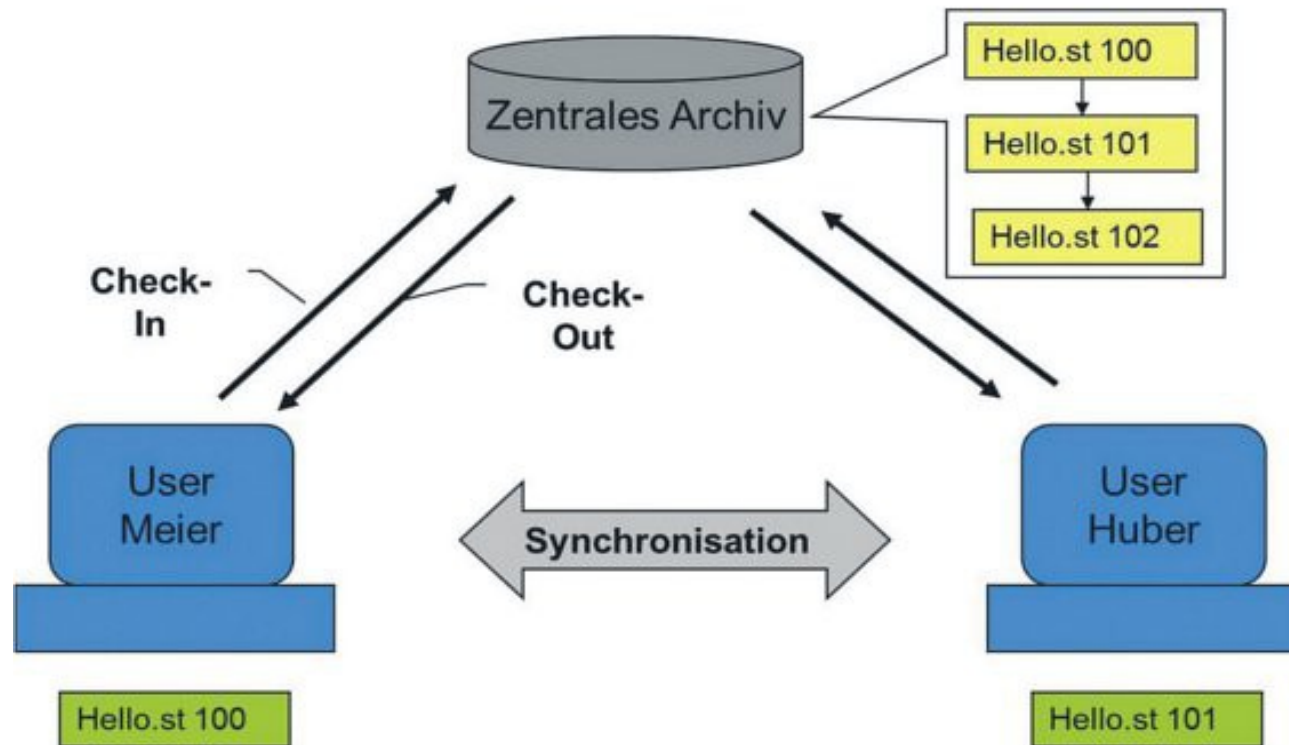
Fachbegriffe

- Repository: Serverbasierte Datenstruktur zur Speicherung von Dateien und Verzeichnissen
- Commit: Festgeschriebener Versionsstand
- Branch: Ein einzelner Entwicklungszweig
- Revision: Fachbegriff für einen bestimmten Entwicklungsstand z. B. der eines Entwicklers
- Update: Synchronisation des Arbeitsverzeichnisses mit dem Repository
- Workspace: Ein Verzeichnis mit einem Repository

Überblick

1. Motivation
2. Fachbegriffe zur Versionierung
- 3. Prinzip der Versionsverwaltung**
4. Arten der Versionsverwaltungen
5. Branching und Merging
6. Versionierungssystem Git
7. GitHub
8. Fazit

Prinzip der Versionsverwaltung



Quelle: <http://www.computer-automation.de/steuerungsebene/steuern-regeln/artikel/79313/1/>

Prinzip der Versionsverwaltung

- Repository ist Anlaufstelle für Entwickler
- Projektdaten werden im Repository abgelegt
- Verbale Erklärung der Abbildung von Folie 8:
 - Jeder Entwickler hat Zugriff auf ein zentrales Repository
 - Vor Arbeitsbeginn: Daten aus Repository laden (Fachbegriff: Auschecken)
 - Nach fertiger Arbeit: Dateien im Repository ablegen (Fachbegriff: Einchecken)

Überblick

1. Motivation
2. Fachbegriffe zur Versionierung
3. Prinzip der Versionsverwaltung
- 4. Arten der Versionsverwaltungen**
5. Branching und Merging
6. Versionierungssystem Git
7. GitHub
8. Fazit

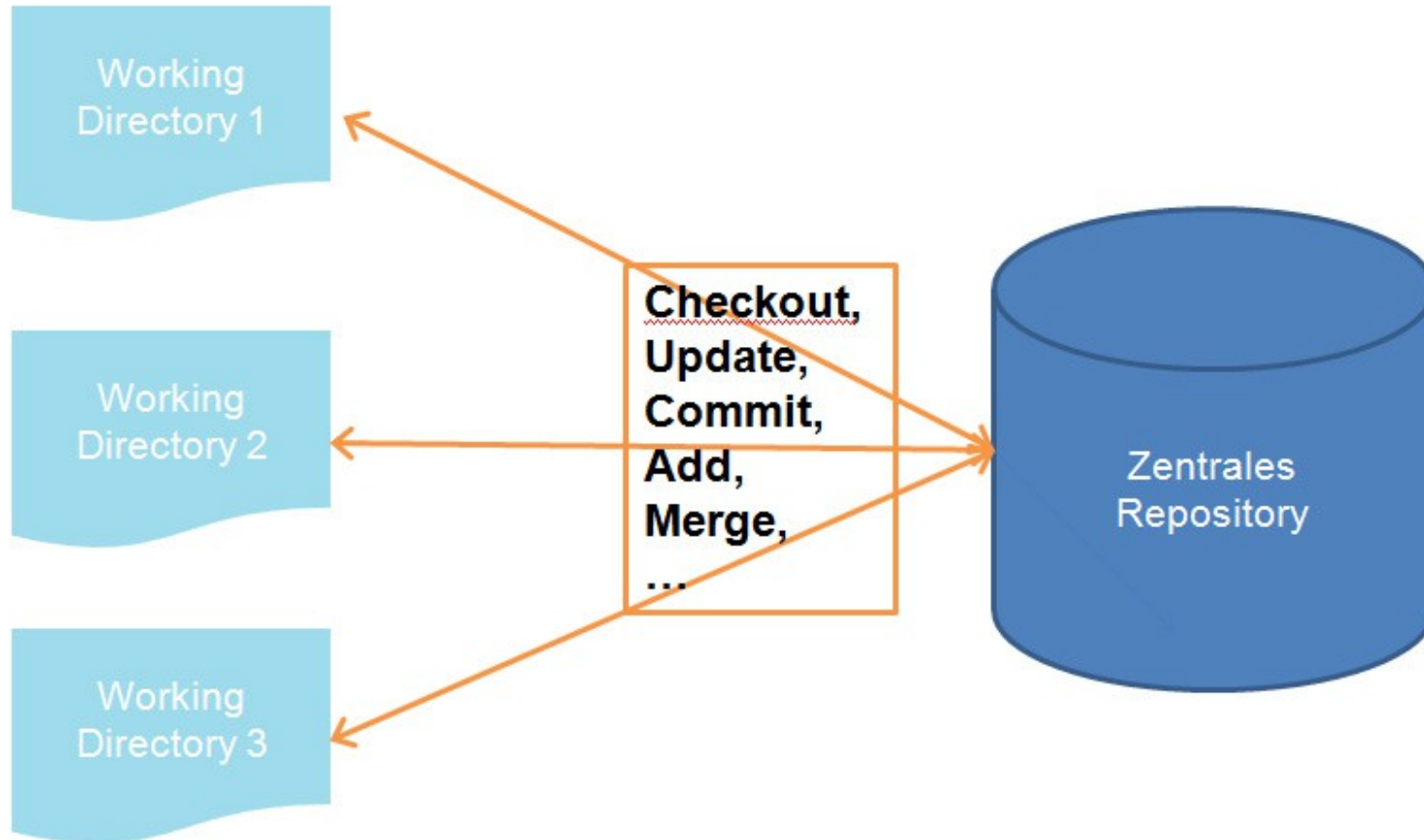
Arten der Versionsverwaltung

- Es gibt 3 Arten von Versionsverwaltungen:
 - Lokale Versionsverwaltung
 - Zentrale Versionsverwaltung
 - Dezentrale Versionsverwaltung
- Hauptunterschiede liegen jeweils in ...
 - der Datenhaltung sowie
 - der Architektur

Lokale Versionsverwaltung

- Sehr einfache und primitive Lösung
- Projekthistorie wird lokal auf dem eigenen Rechner angelegt und verwaltet
- Findet in der Praxis keine Anwendung
- Kontraproduktiv für erfolgreiche Teamarbeit
- Nachteil: Falls Festplatte des Rechners ausfällt, sind alle Projektdaten verloren

Zentrale Versionsverwaltung

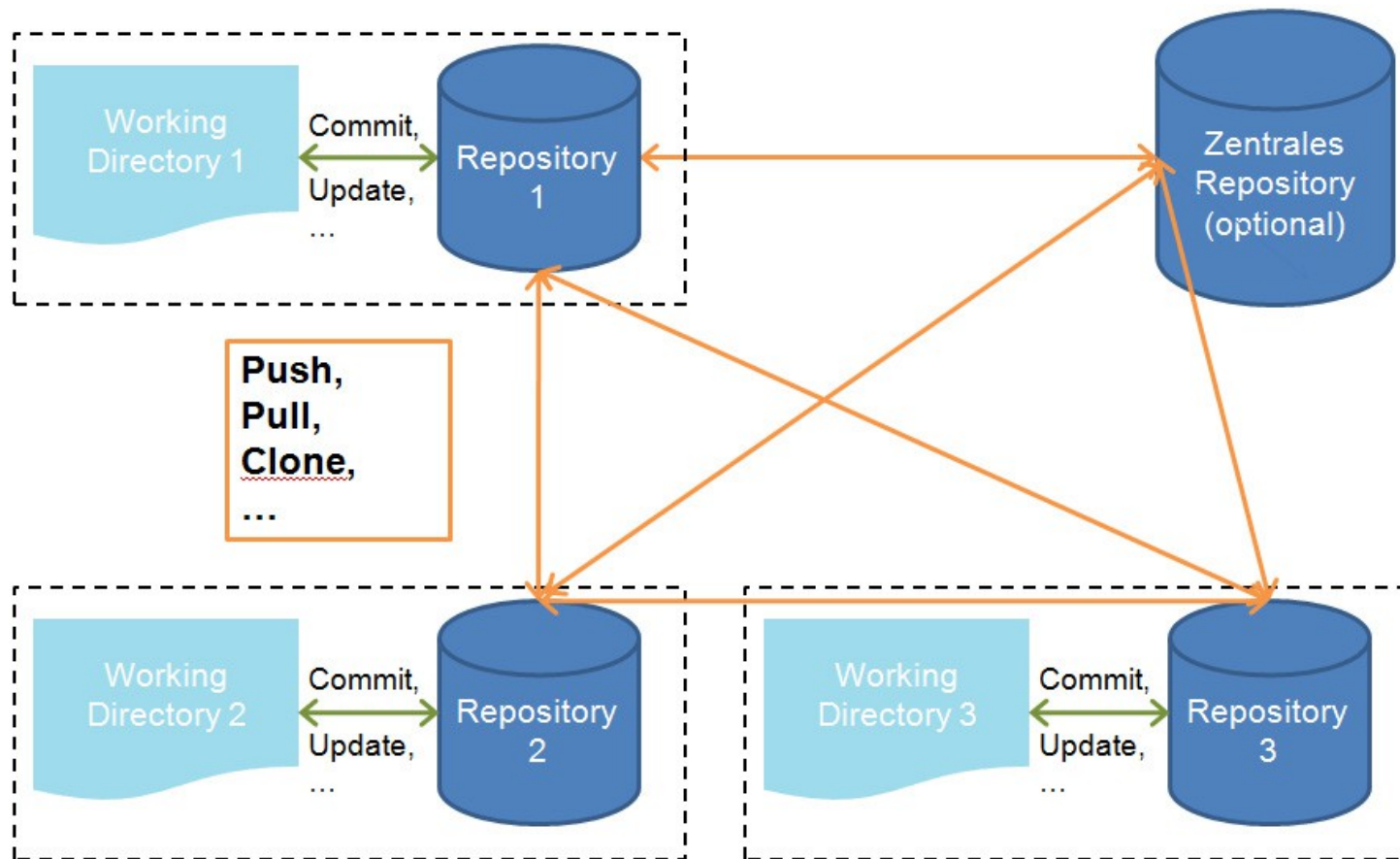


Quelle: <http://www.fh-wedel.de/~si/seminare/ws08/Ausarbeitung/06.vvw/vvw1.htm>

Zentrale Versionsverwaltung

- Sehr einfache und primitive Lösung
- Projekthistorie wird lokal auf dem eigenen Rechner angelegt und verwaltet
- Sehr populäre Lösung bei Projekten
- Namhafter Vertreter sind
 - Subversion
 - Bazaar
- Nachteil: Internetverbindung notwendig

Dezentrale Versionsverwaltung



Quelle: <http://www.fh-wedel.de/~si/seminare/ws08/Ausarbeitung/06.vww/vww1.htm>

Dezentrale Versionsverwaltung

- Jedes Teammitglied hat Arbeitsverzeichnis mit Projektdateien und lokales Repository
- Unterschiede zur zentralen Verwaltung:
 - Existenz eines lokalen Repositories
 - Änderungen landen zuerst im lokalen Repository und werden später in ein Austausch-Repository übertragen
 - Vorteil: Keine Internetanbindung nötig
- Git ist ein dezentrales Verwaltungstool

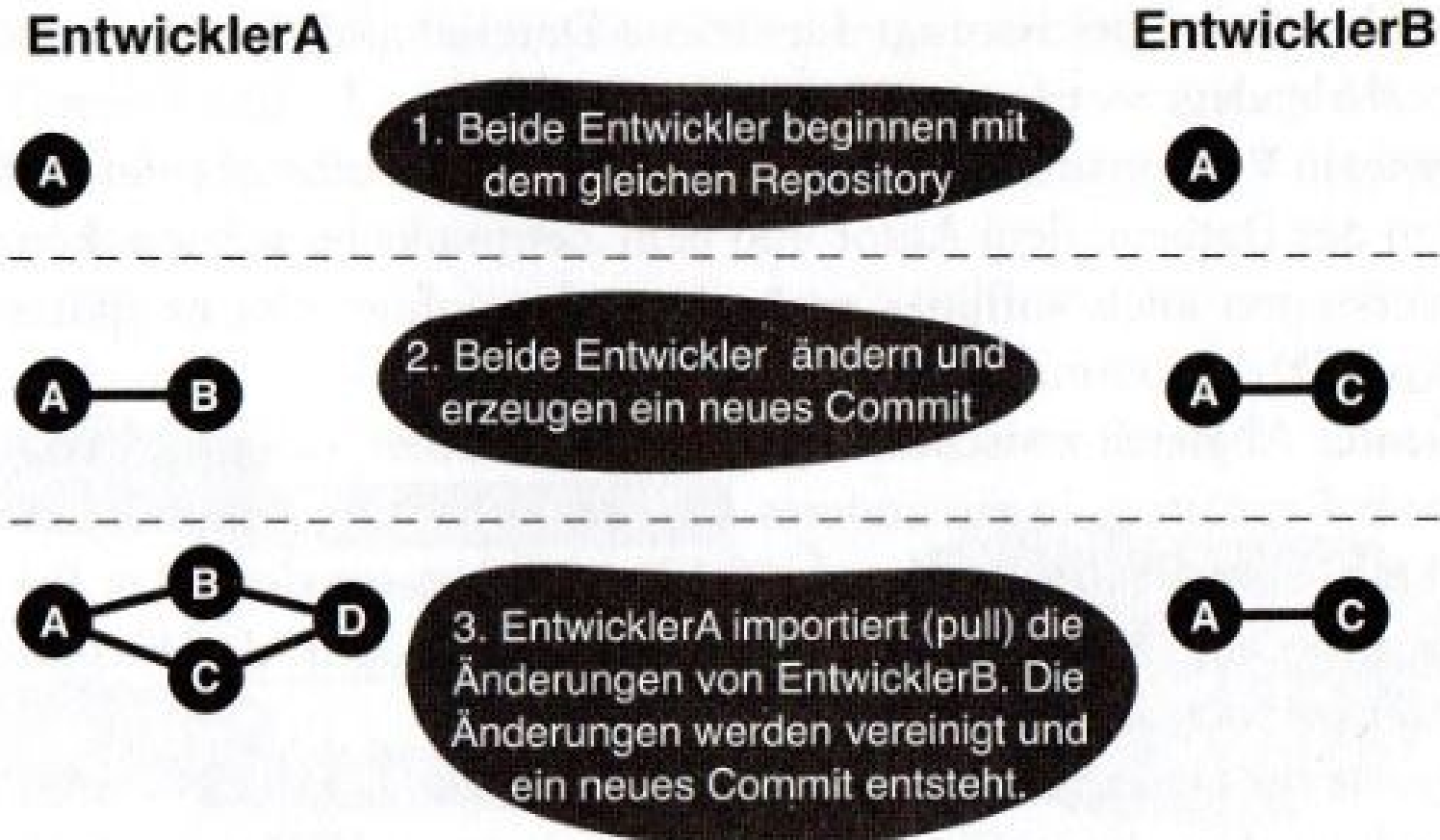
Überblick

1. Motivation
2. Fachbegriffe zur Versionierung
3. Prinzip der Versionsverwaltung
4. Arten der Versionsverwaltungen
- 5. Branching und Merging**
6. Versionierungssystem Git
7. GitHub
8. Fazit

Branching und Merging

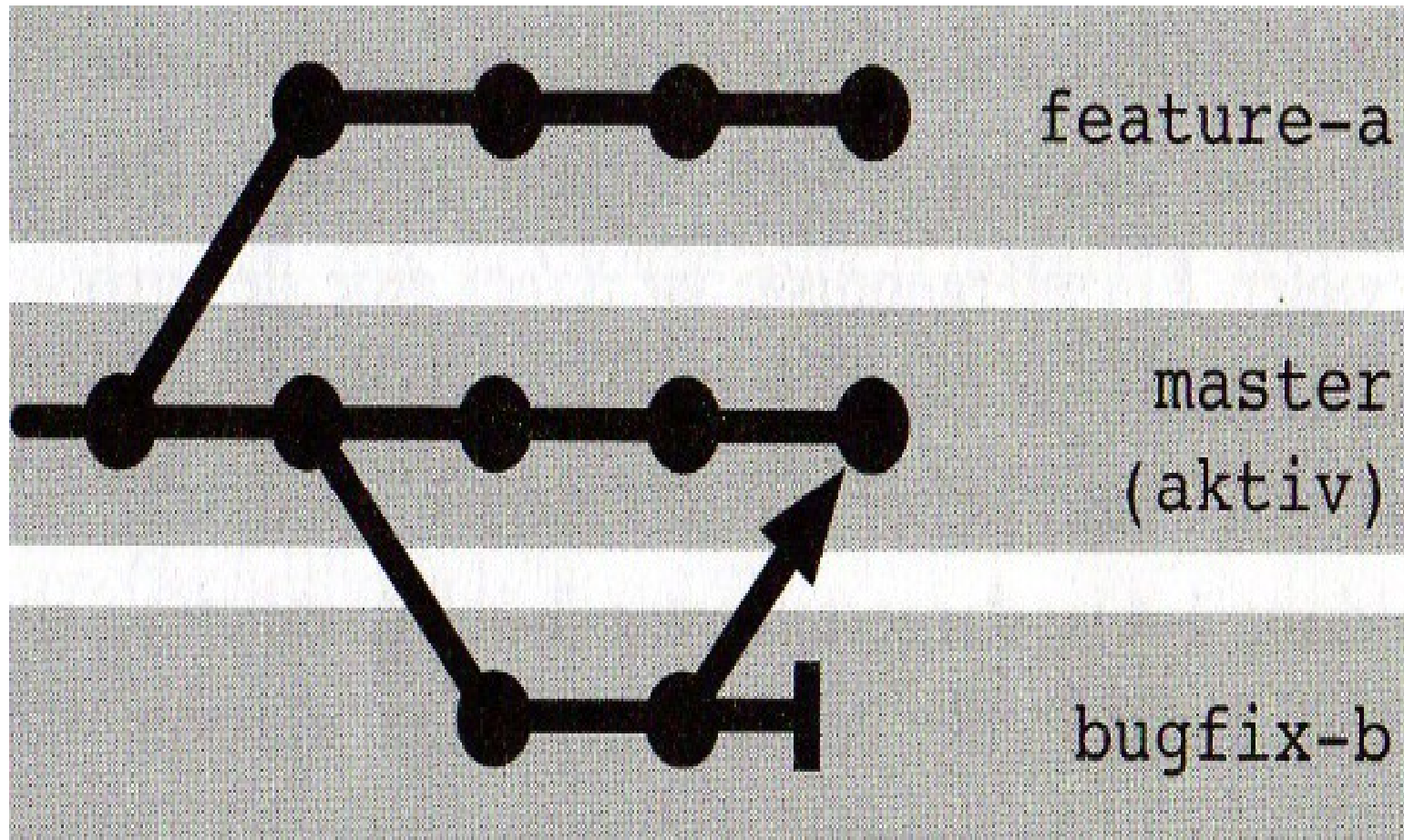
- Branches entstehen durch (gleichzeitiges) paralleles Arbeiten am selben Branch (Illustration siehe Folie 19)
 - Beispiel: Drei Entwickler nehmen Änderungen ausgehend von einem Basisstand (Master-Branch) vor
 - Folge: Drei Branches entstehen und müssen zusammengeführt werden
 - Die Zusammenführung erfolgt innerhalb eines Commits (Stichwort Merge-Commit)
 - Nach der Zusammenführung sind alle Entwickler auf dem gleichen Stand

Illustration Branching und Merging



Quelle: Preißel, René/Stachmann, Bjørn: Git. 2012, S. 6

Isolation von Änderungen/Features



Quelle: Preißel, René/Stachmann, Bjørn: Git. 2012, S. 6

Überblick

1. Motivation
2. Fachbegriffe zur Versionierung
3. Prinzip der Versionsverwaltung
4. Arten der Versionsverwaltungen
5. Branching und Merging
- 6. Versionierungssystem Git**
7. GitHub
8. Fazit

Das Versionierungssystem Git

- Dezentrales Versionierungssystem
- Steht kostenlos zur Verfügung
- Download-Link: <http://git-scm.com/downloads>
- 2005 von Linus Torvalds geschrieben
- Torvalds Intention: Einfachere und sichere Entwicklung des Linux-Kernels ermöglichen
- Durchführung von Branching und Merging gestaltet sich sehr einfach und sicher

Benutzung von Git

- Bedienung über Kommandozeile
- Nutzung grafischer Oberflächen
 - gitk
 - git-gui
 - Qgit
 - gitg
 - Gigggle

Einrichtung

- Bevor die produktive Arbeit beginnt, muss Git mitgeteilt werden, wer man ist
- Zweck: Zur Identifikation des Autors einer Revision
- config-Befehl eignet sich zur Erstellung einer neuen Identität
- Identifikation mit Name und E-Mail (siehe unten)

```
git config --global user.name NAME  
git config --global user.email EMAIL@ADRESSE.de
```


Wichtige Befehle für Git

Befehl	Kurzbeschreibung
git init	Erstellt ein neues Git-Repository
git add [DATEIEN]	Fügt Dateien dem nächsten stattfindenden Commit hinzu
git rm [DATEIEN]	Markiert Dateien als gelöscht für das nächste stattfindende Commit
git status	Zeigt Änderungen seit dem letzten Commit
git log	Zeigt die Historie des Projekts an
git clone	Kopiert das Arbeitsverzeichnis eines Entwicklers inkl. enthaltenem Repository
git pull	Holt Änderungen anderer Entwickler ab und aktualisiert die eigene lokale Revision
git push	Überträgt ein neues Commit ins zentrale Repository (Austausch-Repository)

Demonstration am Rechner



Überblick

1. Motivation
2. Fachbegriffe zur Versionierung
3. Prinzip der Versionsverwaltung
4. Arten der Versionsverwaltungen
5. Branching und Merging
6. Versionierungssystem Git
- 7. GitHub**
8. Fazit

GitHub

- Gründung erfolgte im Februar 2008
- Hostingdienst für Softwareentwicklungsprojekte
- Erfreut sich großer Beliebtheit
- Anlegen von Repositories möglich
 - Öffentliche Repositories
 - Private Repositories mit geschütztem Zugriff
 - Private Repositories sind kostenpflichtig
- Derzeit 3 Millionen Nutzer (Stand: Januar 2013)

Features von GitHub

- Grafische Darstellung aller Aktivitäten einzelner Entwickler
- SSH-Zugang sowie HTTPS-Zugang möglich
- Einsehen der Inhalte von öffentlichen Repositories
- Inhalte öffentlicher Repositories herunterladbar
- Network graph: Einsicht aller Branches und Abspaltungen anhand einer Zeitachse
- Commit-Hooks: Bestimmte Aktionen nach Commits wie z. B. E-Mail-Benachrichtigung an Kollegen

Überblick

1. Motivation
2. Fachbegriffe zur Versionierung
3. Prinzip der Versionsverwaltung
4. Arten der Versionsverwaltungen
5. Branching und Merging
6. Versionierungssystem Git
- 7. Fazit**

Fazit

- Schnelles und effizientes Arbeiten möglich
- Sehr ausgereifter Eindruck
- Kann kostenlos heruntergeladen werden
- Spielt immer größere Rolle in Projekten
- Vorteil: Keine Internetverbindung notwendig
- Verteilte Versionsverwaltungen sind die Zukunft
 - Marktanteile von Git & Co werden steigen
 - Hype um Git könnte Tod anderer Produkte bedeuten

Ende

Vielen Dank für die Aufmerksamkeit!

Noch Fragen?