

Versionsmanagement

1. Grundlagen

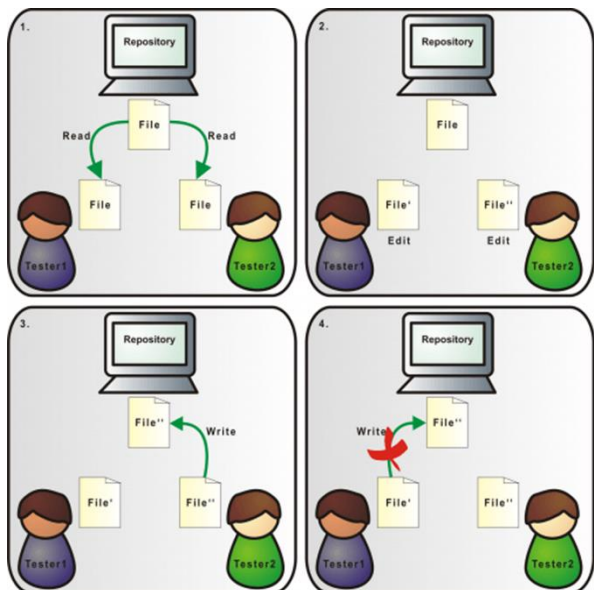
1.1 Wozu ein Versionsmanagement?

- Protokollierung von Änderungen
- Archivierung einzelner Stände
- Wiederherstellung alter Stände
- Koordinierung des Zugriffs von mehreren Entwicklern

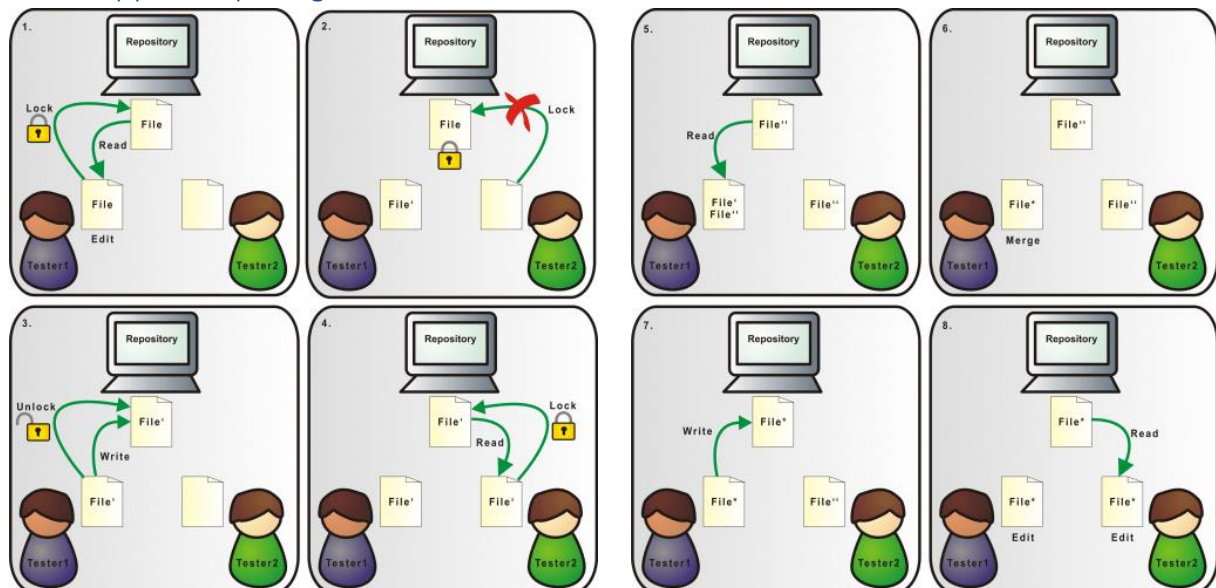
1.2 Konzepte

1.2.1 Lock-Modify-Unlock

- Datei wird vor dem Bearbeiten vom Nutzer gesperrt und kann von keinem anderen Nutzer bearbeitet werden
- Datei im Repository wird mit der geänderten Datei ersetzt



1.2.2 Copy-Modify-Merge



- gleichzeitige Dateiänderung möglich
- anschließendes Zusammenführen der Dateien

1.3 Grundlegende Begriffe

- Repository: digital verwaltetes Projektarchiv
- Working Directory: lokales Arbeitsverzeichnis eines Benutzers
- Revision: Definierter Entwicklungsstand einer Software
- Commit/Check-In: Übertragung der Daten aus dem Working-Directory in das Repository (erzeugt eine neue Revision)
- Update: Übertragung der Daten aus dem Repository in das Working-Directory
- Fork: Aufspaltung des Hauptprojekts in mehrere Entwicklungszweige, die später wieder zusammengeführt werden oder unabhängig voneinander weiterentwickelt werden.
- Branch: Entwicklungszweig einer Version, der es ermöglicht unterschiedliche Versionen einer Software parallel zu entwickeln. Der Hauptentwicklungszweig wird als Trunk/Master bezeichnet.
- Merge: Vereinigung von Entwicklungszweigen zu einem Branch
- Tag: frei wählbarer Bezeichner, der eine bestimmte Revision kennzeichnet (z.B. Versionsnummer)
- Change Set: Menge an Änderungen z.B. zwischen zwei Revisionen

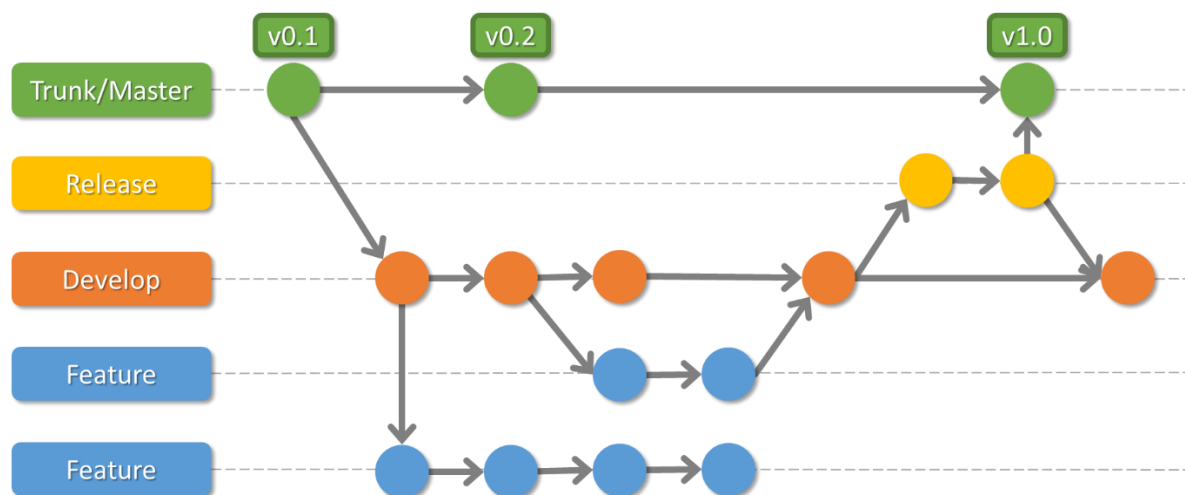


Abbildung 1: Beispiel für Branches im Softwareentwicklungsprozess

2. Verschiedene Repository-Systeme

2.1 lokale Systeme

- Meistens nur eine Datei
- Speichert Version in der Datei des Dokuments
- Verwendung meistens in Büroanwendungen

2.2 Zentrale Systeme

- Client-Server-System
- Zentrales Repository
- Zugriff auf Repository über Netzwerk möglich
- Rechteverwaltung
- Änderungen zwischen den Versionen werden gespeichert
- Tools: CVS, Subversion

2.3 Verteilte Systeme

- Jeder Projektteilnehmer hat eigenes Repository
- Abgleich mit anderen Repositories möglich
- Änderungen sind lokal verfolgbar
- Netzwerkverbindung nicht nötig
- Zentrales Repository möglich aber nicht nötig
- Tools: Git, Mercurial

2.4 Vergleich

Zentrale Systeme	Verteilte Systeme
zentrales Repository	zentrales Repository möglich
Rechteverwaltung	keine Rechteverwaltung
beide Prinzipien nutzbar	Copy Modify Merge-Prinzip
Änderungen zwischen den Versionen werden gespeichert	Daten werden als Snapshots gespeichert
Server notwendig (da Client-Server-System)	Server nicht notwendig