

# ) Seminar )

## Versionsverwaltung mit Git

Bei der praktischen Anwendung eines verteilten Versionsverwaltungssystems wie git wird schnell klar, dass nicht nur die zur Verfügung stehenden Kommandos und Funktionen neu sind. Dieser Bruch mit althergebrachten Konzepten eröffnet vor allem in den Bereichen Workflows und Best Practices neue Ansätze.

Das Seminar gibt eine Einführung und stellt den Teilnehmern die Git typischen Arbeitsweisen vor. Außerdem wird gezeigt, wie eigene Workflows mit Git frei definiert werden können, welche Best Practices sich hierbei im Enterprise Umfeld bewährt haben und wie die bereits existierende Arbeitsweisen sinnvoll auf Git übertragen werden können.

### Seminar-Inhalt

#### Einführung in die Versionsverwaltung mit Git

- Entstehung von Git im Kontext der Linux Kernel-Entwicklung
- Grundbegriffe: Repository, Commit, Tag, Branch und Merge
- Bedeutung des Releasemanagements
- Vergleich mit einem zentralen Ansatz am Beispiel Subversion

#### Versionierung von Dateien mit Git

- Erstellung eines Repositories (Init)
- Grundlegender Versionsverwaltungs-Workflow (Add, Commit)
- Änderungen untersuchen (Log)
- Erstellung von Entwicklungszweigen (Branches, Merge) und Versionen (Tags)

#### Fortgeschrittene Themen

- Die Kommunikation zwischen Repositories (Clone, Fetch, Pull, Push)
- Umsetzung eines zentralen Repositories (Bare Repository)
- Erweiterte Git Funktionen (Rebase, Cherry-Pick, Stash, Reflog, Diff)
- Ändern/Änderungen rückgängig machen (Amend Commit, Revert)
- Zurücksetzen des Arbeitsbereichs und von Entwicklungszweigen (Reset)
- Wiederkehrende Schemata verschiedener Git Kommandos (Refspecs, Version Ranges, ...)

#### Konzepte und Tooling

- Git und CVS/Subversion
- Überblick gängiger graphischer Clients (Git GUI, Atlassian SourceTree, Eclipse EGit, TortoiseGit)
- Möglichkeiten zum Repository Hosting (GitHub, Bitbucket, Atlassian Bitbucket, Gitblit, Gitolite, ...)
- Server-Zugriff per http/https/SSH

#### Best Practices und Begrifflichkeiten

- Separation of concerns
- Commit early & commit often
- Topic-Branches
- Merge vs. Rebase
- Up vs Down-Merge
- Fork vs Branch
- Pull Requests und deren Verwaltung (Stichwort Gatekeeper)
- Code Reviews in den Workflow integrieren

#### Gitflow als Branching- und Workflow-Konzept

- Aufgaben von Entwicklungszweigen (Branching Modell)
- Vorgehen und Workflow bei der Umsetzung von Änderungen
- Tool support (z.B. in SourceTree)

#### Erweiterte Konzepte und Optimierungen

- Verschachteln von Repositories (Submodules)
- Strukturierung von Repositories und deren Optimierung
- Shallow Clones
- Sparse Checkouts
- Git Repository Internals
- Merge Strategien

### Zielgruppe

Entwickler, die die Vorteile einer verteilten Versionsverwaltung im Projektalltag nutzen möchten.

### Voraussetzungen für Seminar-Teilnahme

PC Grundkenntnisse. Erfahrungen mit anderen Werkzeugen zur Versionsverwaltung wie z.B. Versionsverwaltung mit SVN oder CVS sind von Vorteil aber keine Voraussetzung.

### Seminar-Dauer

2 Tage

### Vorteile einer Seminar-Teilnahme

- Sichern Sie den Erfolg Ihrer Softwareprojekte durch den Einsatz des verteilten SCM Systems Git
- Die Schulung vermittelt die typischen Arbeitsabläufe von Git anhand vieler Übungen und Beispiele
- Lernen Sie wie man typische Workflows im Kontext einer verteilten Versionsverwaltung einsetzt
- Unsere Kurs-Unterlagen sind immer auf dem aktuellsten Stand

### Seminar-Preis

1.255 EUR pro Person  
(inkl. Unterlagen u. Tagesverpflegung zzgl. MwSt.)

### Seminar-Termine

Termine gerne auf Anfrage möglich.

### Inhouse-Kurse

Alle unsere Seminare können wir Ihnen auch für eine Durchführung in Ihrem Hause mit einer speziellen Kalkulation für Exklusiv-Seminare anbieten. Hierbei können die Inhalte aller unserer Seminare beliebig für Ihr individuelles Training zu einem Wunschseminar zusammengestellt werden.