

# Continuous Integration, Deployment und Delivery mit Subversion / SVN, Maven und Jenkins (S1716)

Mit Continuous Integration (Kontinuierliche Integration) wird die Qualität einer Software im Entwicklungsprozess sichergestellt. Continuous Integration definiert einen Prozess, in dem die Bausteine Versionsverwaltung, Übersetzen, Kompilieren und Linken (Build-Prozess) sowie automatisierte Tests und Erstellung von Softwaremetriken „integriert“ sind. Durch das „Einchecken“ von Softwarecode in die Versionsverwaltung wird der gesamte Vorgang automatisch ausgelöst. Diese Veranstaltung zeigt, wie man mit Subversion (SVN), Maven und Jenkins ein Continuous Integration-System aufbaut.

## Offene Termine

Preis	Dauer
Auf Anfrage	5 Tage. Die vorgeschlagene Dauer kann Ihren Bedürfnissen angepasst werden

### \* Buchen ohne Risiko

- › Keine Vorkasse
- › Kostenloses Storno bis zum Vortag des Seminars
- › Rechnung nach erfolgreichem Seminar

✓ Garantierter Termin und Veranstaltungsort

€ Preise zzgl. Mehrwertsteuer

3=2 Der dritte Mitarbeiter nimmt kostenlos teil

## Weitere Buchungsmöglichkeiten

<b>Firmenschulung</b>	Schulung für Ihre Mitarbeiter mit individuellen Inhalten zum Wunschtermin im GFU-Schulungszentrum.
<b>Inhouse-Schulung</b>	Schulung für Ihre Mitarbeiter mit individuellen Inhalten zum Wunschtermin in Ihrem Hause.
<b>Individualschulung</b>	Schulung für eine Einzelperson mit individuellen Inhalten zum Wunschtermin, wahlweise in Ihrem Hause oder im GFU-Schulungszentrum.

## Schulungs-Ziel

Sie verstehen, was Subversion ist und wie Sie es erfolgreich zur Koordination Ihrer Artefakte in Ihrem Software-Projekt einsetzen können. Sie können Softwareprojekte mit Maven "bauen". Sie lernen, Jenkins in Ihrem Projekt effektiv zu nutzen.

## Inhalt

- › **Subversion / SVN Einführung**
  - › Basisinformationen zur Architektur für ein besseres Verständnis
  - › Anlegen von Benutzern und deren Zugriffsrechten
- › **Frontends für Subversion / SVN: Tortoise, Eclipse, Visual-Studio**
- › **Ein Repository anlegen mit Subversion/ SVN**
- › **Ein Projekt importieren mit Subversion / SVN**

## Wer sollte teilnehmen

---

Entwickler, Webentwickler, Entwicklerteams und selbständige Entwickler mit Kenntnissen in einer Hochsprache und XML, die auf Basis von Subversion / SVN, Maven und Jenkins eine CI (Continuous Integration) aufbauen wollen.

## Organisation

---

### Teilnehmerzahl

min. 1, max. 8 Personen

### Seminarzeiten

5 Tage, 1. Tag 10:00 - 17:00 Uhr, Folgetage 09:00 - 16:00 Uhr

### Ort der Schulung

GFU-Schulungszentrum Köln oder bei Ihnen als Inhouse-Schulung

## Enthaltene Leistungen

---

### Im Preis enthalten:

- > Voll ausgestatteter Arbeitsplatz pro Teilnehmer
- > Fachbuch zum Seminar
- > Teilnahmezertifikat
- > Kostenloser persönlicher Parkplatz
- > Kostenloser Shuttle-Service
- > Frühstück, Snacks und Getränke ganztägig
- > Mittagessen im eigenen Restaurant, täglich 6 Menüs, auch vegetarisch

## Haben Sie Fragen?

---

Gerne beraten wir Sie persönlich per [Mail](mailto:info@gfu.net) oder Telefon.

- > [info@gfu.net](mailto:info@gfu.net)
- > Infoline 0221 82 80 90

- > **Auschecken und Einchecken mit Subversion / SVN - ein typischer Arbeitszyklus**
- > **Merge-Konflikte lösen mit Subversion / SVN**
- > **Auf alte Versionen zurücksetzen**
- > **Mit Tags einen Stand einfrieren**
- > **Branches in Subversion / SVN erzeugen**
  - > Branches zur Hauptentwicklung zurückführen
- > **Zugriffe mit Ant oder Maven automatisieren**
  
- > **Maven Grundlagen**
  - > Projektstruktur
  - > Konventionen
  - > Projektobjektmodell
  - > Deklaration der Abhängigkeiten und Scope
  - > Maven Lifecycle
  - > Lokales und zentrales Maven Repository
- > **Wichtige Maven Plugins**
  - > Surefire Plugin für JUnit
  - > Compiler Plugin
- > **Konfiguration von Maven**
  - > Maven settings.xml für benutzerspezifische Einstellungen
  - > Maven Properties
  - > Semantic Versioning
  - > SNAPSHOT und Release Versionen
- > **Der Site Lifecycle von Maven**
  - > Surefire Report
  - > ToDo-Liste
  - > Javadoc Report
  
- > **Erweiterte Grundlagen von Maven**
  - > Profile
  - > Proxy-Konfiguration in der settings.xml
  - > Passwörter verschlüsseln
  - > Ersetzen von Schlüsselwörtern in Dateien (Filter)
- > **Multi-Module-Projekte mit Maven**
- > **Standardkonfiguration mit Parent-POM**
  - > Integration von Plugins für die Code-Generierung
  - > Zugriff von Maven auf Versionsverwaltungen (CVS, SVN, git, ClearCase, Mercurial, ...)
- > **Release-Erstellung mit Maven**
  - > Release Plugin
  - > Assembly Plugin
  - > Maven Deployment
- > **Reporting in Maven**
  - > Crossreferenzen mit JXR
  - > UML-Klassendiagramme mit JavaDoc
- > **Reports in Maven für Qualitätssicherung**
  - > Code-Metriken mit Checkstyle, Findbugs, PMD und JDepend
  - > Dynamische Code-Analyse mit Cobertura oder

- JaCoCo
- > Konfiguration der Plugins
- > **Nexus als Firmen-Repository**

#### 4. Tag: Auch separat buchbar als [Continuous Integration mit Jenkins](#)

- > **CI Grundlagen**
- > **Jenkins Grundlagen**
  - > Installation
  - > Konfiguration
- > **Arbeiten mit Jenkins**
  - > Jobs erstellen mit Ant, Maven
  - > Anbindung an SCM wie CVS, SVN, git, ...
  - > Build-Trigger
  - > Plugins für Projektmetriken wie beispielsweise Findbugs und Checkstyle
- > **Jenkins für große Projekte**
  - > Parallele Builds
  - > Build-Reihenfolge
  - > Tuning
- > **Interessante Plugins für Jenkins**
  - > Continuous Delivery mit Build Pipeline Plugin
  - > Parameter und Bedingungen
  - > Fork/Join
- > **Build Jobs für Nicht-Java-Projekte**
  - > C#, make, ...

#### 5. Tag: Auch separat buchbar als [Continuous Deployment und Delivery mit Jenkins](#)

- > **Grundlagen zu Continuous Deployment und Delivery**
  - > DevOps
  - > Unterschiede zwischen Continuous Deployment und Delivery
  - > Feature Toggle
  - > Branches by Abstraction
- > **Deployment Pipeline (Continuous Deployment)**
  - > Welche Umgebungen brauche ich?
  - > Aufbau einer Deployment Pipeline
  - > Automatisierte ReleasePipelines
  - > Promotion von Builds
  - > Hilfreiche Jenkins Plugins
- > **Test-Automatisierung (Continuous Testing)**
  - > Automatisierte Tests
  - > Jenkins Plugins für Unit, Integration, Functional, NonFunctional, SmokeTests
- > **Umgebungen automatisiert aufbauen**

- > Infrastructure as a Code
- > Provisioning von virtuellen Maschinen mit Docker, Vagrant, Puppet, Chef, Packer o.a.
- > Docker Images mit Maven erstellen
- > Docker Container mit Jenkins starten
- > MatrixBuilds mit Jenkins z.B. für unterschiedliche Datenbanken oder Java Versionen
- > **Ausblick**
  - > Continuous Monitoring (Cycle Time)
  - > Integration in Application Lifecycle Management (ALM) Arbeitsabläufe