



JAVACREAM

*Training
Consulting
Projectmanagement*

GIT

- Name
- Rolle im Unternehmen
- Themenbezogene Vorkenntnisse
- Aktuelle Problemstellung
- Konkrete individuelle Zielsetzung

Einführung

- Quellcode = “Werke” werden vom Versionsverwaltungssystem mit Meta-Informationen ergänzt
 - Wer hat wann warum welche Änderungen gemacht?
- Der Stand eines Projekts wird durch ein Aggregat von Werken definiert
- Zentrale Ablage aller relevanten Informationen
 - inklusive Authentifizierung und Autorisierung
- Werkzeugen und Verfahren für die Team-Zusammenzusammenarbeit
- Werkzeuge für die Visualisierung und Konsolidierung der Arbeit

- Quellcode = “Werke” werden vom Versionsverwaltungssystem mit Meta-Informationen ergänzt
 - Wer hat wann warum welche Änderungen gemacht?
- Der Stand eines Projekts wird durch ein Aggregat von Werken definiert
- Zentrale Ablage aller relevanten Informationen
 - inklusive Authentifizierung und Autorisierung
- Werkzeugen und Verfahren für die Team-Zusammenzusammenarbeit
 - Konzepte und Technik
- Werkzeuge für die Visualisierung und Konsolidierung der Arbeit
 - Konsolenbasiert

- Quellcode = “Werke” werden vom Versionsverwaltungssystem mit Meta-Informationen ergänzt
 - Wer hat wann warum welche Änderungen gemacht?
- Der Stand eines Projekts wird durch ein Aggregat von Werken definiert
- Zentrale Ablage aller relevanten Informationen
 - inklusive Authentifizierung und Autorisierung
 - Produkt-Lösungen, z.B. **GitHub** (Microsoft), **GitLab** (GitLab.com), **BitBucket** (Atlassian)
- Werkzeugen und Verfahren für die Team-Zusammenzusammenarbeit
 - Konzepte und Technik
 - **Pull / Merge Requests**
- Werkzeuge für die Visualisierung und Konsolidierung der Arbeit
 - Konsolenbasiert
 - Web Frontend



- Quellcode = “Werke” werden vom Versionsverwaltungssystem mit Meta-Informationen ergänzt
 - Wer hat wann warum welche Änderungen gemacht?
- Der Stand eines Projekts wird durch ein Aggregat von Werken definiert
- Zentrale Ablage aller relevanten Informationen
 - inklusive Authentifizierung und Autorisierung
 - Produkt-Lösungen, z.B. **GitHub** (Microsoft), GitLab (GitLab.com), BitBucket (Atlassian)
- Werkzeugen und Verfahren für die Team-Zusammenzusammenarbeit
 - Konzepte und Technik
 - **Pull / Merge Requests**
- Werkzeuge für die Visualisierung und Konsolidierung der Arbeit
 - Konsolenbasiert
 - Web Frontend
 - Entwicklungsumgebungen (IntelliJ, Visual Studio Code) native-Git-Installationen mit Tortoise, ...



- Quellcode = “Werke” werden vom Versionsverwaltungssystem mit Meta-Informationen ergänzt
 - Wer hat wann warum welche Änderungen gemacht?
- Der Stand eines Projekts wird durch ein Aggregat von Werken definiert
- Zentrale Ablage aller relevanten Informationen
 - inklusive Authentifizierung und Autorisierung
 - Produkt-Lösungen, z.B. **GitHub** (Microsoft), GitLab (GitLab.com), BitBucket (Atlassian)
- Werkzeugen und Verfahren für die Team-Zusammenzusammenarbeit
 - Konzepte und Technik
 - **Pull / Merge Requests**
- Werkzeuge für die Visualisierung und Konsolidierung der Arbeit
 - Konsolenbasiert
 - Web Frontend
 - Entwicklungsumgebungen (IntelliJ, Visual Studio Code) native-Git-Installationen mit Turtoise, ...

Seminar
Tag 1 +x

Seminar
Tag 2

First Contact

- Installiert wird das Executable “git”
 - Kein Client, der mit einem Server kommuniziert, sondern das komplette Core-Versionsverwaltungssystem
 - Kein Hintergrund-Dienst, Service, Daemon
 - Das Git-Executable stellt während der Kommando-Ausführung die Funktionalität eines Versionsverwaltungssystems zur Verfügung

- ~~Server-Administrator legt einen Account für Sie an~~

- ~~Konfiguration: Server-URL~~

- Erstellung einer Git-Konfigurationsdatei (eine einfache Text-Datei namens .gitconfig in Ihrem User-Home)

- Minimal:

- user.name

- user.email

- git config --global user.name "Rainer Sawitzki"

- git config --global user.email training@rainer-sawitzki.de

-

Git Core braucht
keine Server