



Werkzeuge

Werkzeuge für den (Java-) Entwickler

- Entwicklungsumgebung
- Test-Tools
- Build-Werkzeuge
- Versionsmanagement

Integrierte Entwicklungsumgebung Eclipse

- Kostenfreies, quelloffenes Programmierwerkzeug zur Entwicklung von Software
- Entwickelt von IBM als Nachfolger von Visual Age für Java, dann zu Open Source Project gemacht.
- Weltweit mit am häufigsten für die Entwicklung von Java-Software eingesetzt
- Sehr modular aufgebaut, vorkonfigurierte Zusammenstellungen werden zum Download angeboten
- Es gibt eine Vielzahl sowohl quelloffener als auch kommerzieller Erweiterungen
- Nicht nur für Java-Entwicklung nutzbar

Integrierte Entwicklungsumgebung IntelliJ

- IDE für die Programmiersprachen Java, Kotlin, Groovy und Scala
- Kommerziell entwickelt und gepflegt von JetBrains
- Aktuell 2 verschiedene Versionen verfügbar:
 - Kostenfreie Open Source Community Edition (beschränkt auf Basis-Funktionalität)
 - Kostenpflichtige Ultimate Edition
- Wird als die „beste“ Entwicklungsumgebung für Java-Entwicklung eingeschätzt

Testen mit JUnit

- Unit- und Integrationstests können durch den Entwickler mit Hilfe des freien Test-Frameworks erfolgen
- Ist heute (hoffentlich) übliche Praxis in der Software-Entwicklung mit Java
- Aktuell 2 syntaktisch etwas unterschiedliche Versionen eingesetzt:
 - JUnit 4
 - JUnit 5, kurz Jupiter genannt
- Freies Konkurrenz-Framework TestNG hat sich weniger stark durchgesetzt
- Ergänzt durch weitere Bibliotheken, z.B. für Mocking

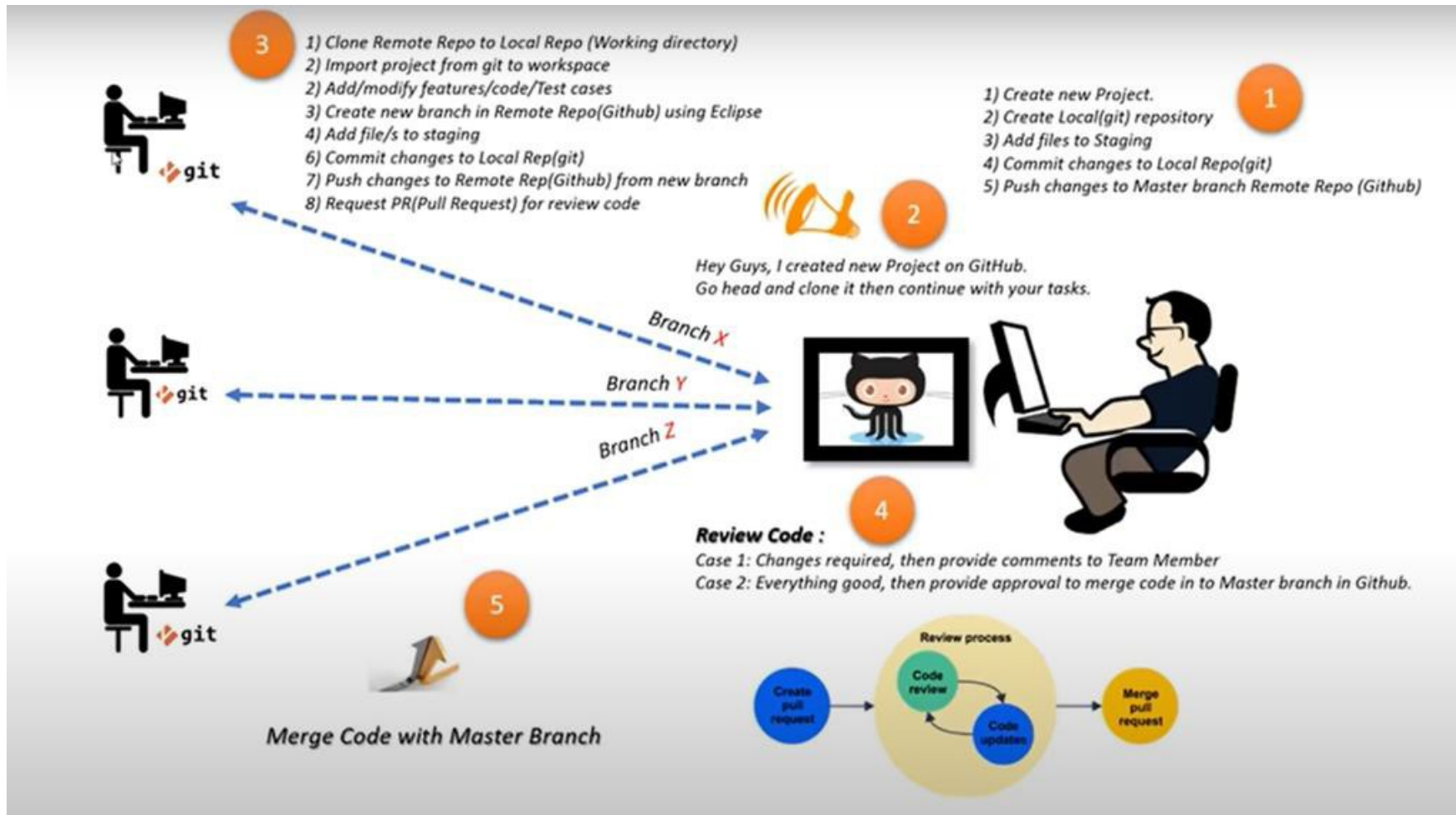
Build-Management mit Maven

- Basiert auf Java und ist an sich Kommandozeilen-Werkzeug (Linux-Welt)
- Buildmanagement ist unabhängig von der Entwicklungsumgebung
 - Bereitstellung der benötigten Bibliotheken in Form von JARs (Dependency-Management)
 - Übersetzen
 - Testen
 - Packen
 - Deployen
- Ziele (Convention over Configuration)
 - Vereinfachung des Build-Prozess
 - Vereinheitlichung der Projektstruktur und des Build-Managements (unabhängig von der IDE)
 - Bereitstellung von Projektinformation
 - Anregung zu verbesserter Entwicklungspraxis
 - Flexibel erweiterbar (Plugin-Mechanismus)
- Maven beinhaltet keine IDE, die wird für die Erstellung des Programmcodes nach wie vor benötigt
 - Maven-Projekte können aber in alle heutigen IDEs integriert werden
- Alternativen zu Maven: IDE-Helferlein, ANT (veraltet), Gradle

Versionskontrolle mit Git

- Versionskontrolle für Dateien, vorrangig für Software-Projekte, die für Teamarbeit quasi unverzichtbar ist
- Git ist verteiltes Versionierungssystem, das als Open Source Software zur Verfügung gestellt wird
- Weltweit häufigstes VCS in der Software-Entwicklung
- Anders als die meisten Vorläufer (Subversion, CVS, Clearcase) nicht auf einen zentralen Server angewiesen, unabhängiges Arbeiten ist möglich
- Wird aber in der Regel in Kombination mit einem zentralem Repository zwecks Kommunikation und Kontrolle bei Teamarbeit eingesetzt
- Es gibt Server-Produkte
 - Verwaltung vieler Repositories
 - Benutzermanagement
 - Einbindung in die IT-Infrastruktur (Ticketsystem, Buildmanagement, Continuous Integration)
 - Kandidaten: GitHub, GitLab, BitBucket

Git Workflow

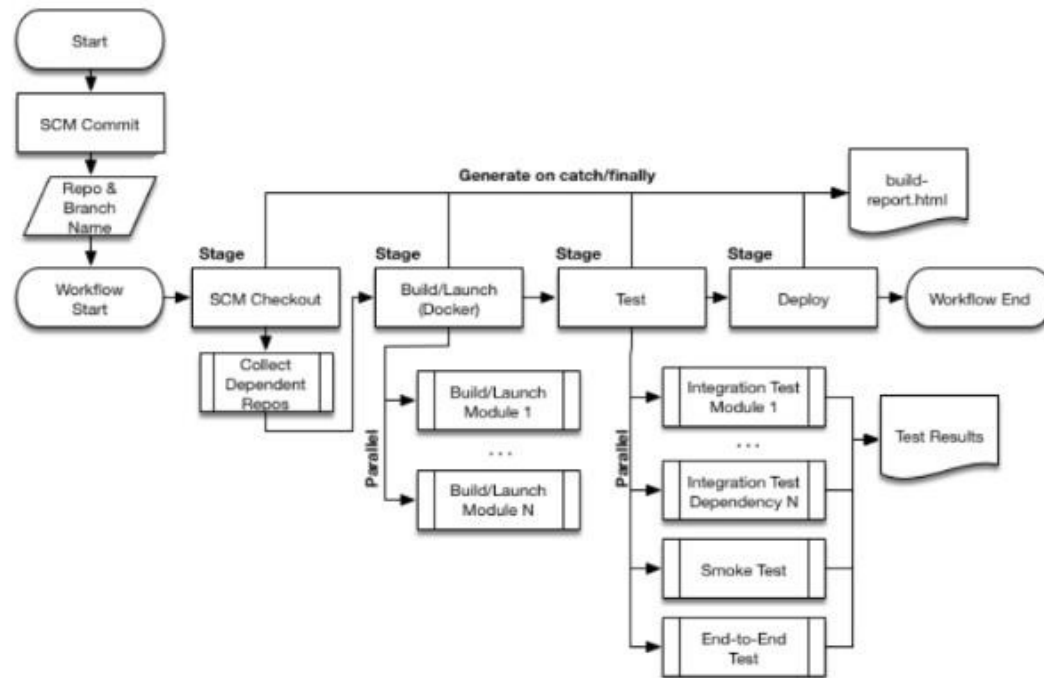


CI - CD





- Wollen wir immer schneller qualitativ hochwertige Software bereit stellen so muss der Softwareentwicklungsprozess immer stärker automatisiert werden
- Eine Delivery-Pipeline modelliert diesen Prozess
- Continuous Deployment , Delivery und Integration sind drei unterschiedliche „Ausbaustufen“
- Integrations- und Release-Management-Tools ermöglichen
 - den Fortschritt jeder Änderung zu sehen und zu kontrollieren, während sie von der Versionskontrolle über verschiedene Test und Bereitstellungsphasen bis zur Freigabe an die Benutzer verläuft.

CD-Pipeline



Tools für die CD-Pipeline

	 Jenkins	 circleci	 TeamCity	 Bamboo	 GitLab
Open source	Yes	No	No	No	No
Ease of use & setup	Medium	Medium	Medium	Medium	Medium
Built-in features	3/5	4/5	4/5	4/5	4/5
Integration	★ ★ ★ ★ ★	★ ★ ★	★ ★ ★ ★	★ ★ ★	★ ★ ★ ★
Hosting	On premise & Cloud	On premise & Cloud	On premise	On premise & Bitbucket as Cloud	On premise & Cloud
Free version	Yes	Yes	Yes	Yes	Yes
Build Agent License Pricing	Free	From \$39 per month	From \$299 one-off payment	From \$10 one-off payment	From \$4 per month per user
Supported OSs	Windows, Linux, macOS, Unix-like OS	Linux or MacOS	Windows, Linux, macOS, Solaris, FreeBSD and more	Windows, Linux, macOS, Solaris	Linux distributions: Ubuntu, Debian, CentOS, Oracle Linux

Top 5 CI/CD tools in 2020

Jenkins: ein Continuous Integration Server

- Jenkins ist ein OpenSource-Produkt was einen Java-Web-Container (z.B. Tomcat) oder aber JEE-Server voraussetzt
- Continuous Integration mit Jenkins
 - Ziehe Code aus Git
 - Build kontrolliert durch Jenkins, z.B. via Maven
 - Tests kontrolliert durch Jenkins
 - Deployment in die Testumgebung