

---

---

April 2007

---



ex|Xcellent  
solutions

Von ANT nach Maven 2



- 1 | Die Historie
- 2 | Was ist Maven 2
- 3 | Was ist Maven 2 nicht
- 4 | Die Konzepte
- 5 | Von ANT nach Maven 2
- 6 | Zusammenfassung
- 7 | Referenzen

## x| ANT

- | Build-Prozesse in der Java-Welt wurden zum ersten mal mit Java-Mitteln automatisiert

## x| Maven 1

- | Wurde initiiert, um den Build-Prozess im **Jakarta Turbine Projekt** zu vereinfachen
- | Umständliche Konfiguration
- | Schlechte Performance

## x| Maven 2

- | Wurde von Grund auf neu konzipiert
- | Einfache Konfiguration
- | Gute Performance
- | Wurde durch die "Ruby on Rails-Community" populär

**x|** Maven 2 etabliert, für den Entwicklungsprozess bewährte Vorgehensweisen, um **Qualität, Verständlichkeit** und **Wiederverwendbarkeit** zu fördern

| Ein Paradigmenwechsel

- Build **Skripte** → Projekt **Konfiguration**
- **Wie** wird etwas erzeugt → **Was** wird erzeugt
- **Erzeugen** von Produkten → **Managen** von Produkten

| Ein Entwickler-Werkzeug

- ANT stellt eine lose Bauklotzsammlung zum Bau von Projekten zur Verfügung
- Maven 2 **konfiguriert Projekte mit objektorientierten Konzepten**

**x|** Maven 2 definiert:

- | Großteile des Entwicklungsprozess

**x|** Maven 2 automatisiert:

- | Kompilieren von Quellcode und Filtern von Ressourcen
- | Die Abhängigkeiten zwischen Projekten und Bibliotheken
- | Die Qualitätssicherung auf Ebene des Quellcodes
- | Komponenten- und Integrationstests
- | Die Erstellung von Releases
- | Die technische Projektdokumentation

- x| Nur ein Build-Werkzeug
- x| Nur ein Werkzeug, um Dokumentation zu erzeugen
- x| Eine Erweiterung von ANT, um z.B. Abhängigkeiten aufzulösen
- x| Eine Sammlung von ANT-Targets

x| POM

x| Artifact

x| Goal und Plugin

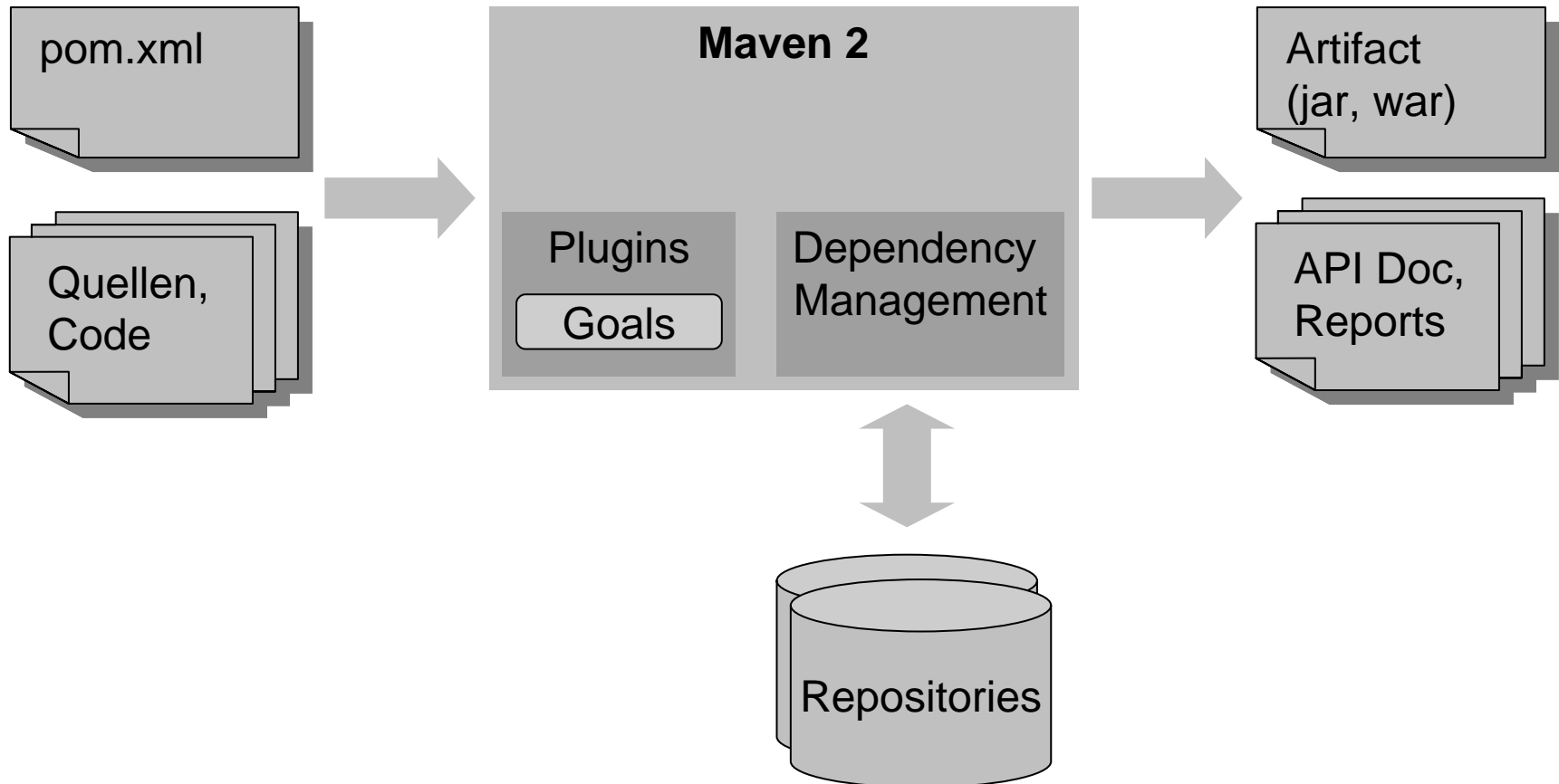
x| Repository

x| Dependency Management

x| Build-Lifecycle

x| Konventionen

x| Projekt-Reports



XML-Datei, welche das Projekt beschreibt:

- | Name, Version und Typ
- | Vererbung
- | Unterprojekte (Module)
- | Abhängigkeiten und Repositories
- | Lifecycle- und Plugin-Konfiguration
- | Allgemeine Informationen (Entwickler, Lizenz, etc.)
- | Issue-Management (Bugzilla, Mailinglisten, etc.)
- | Sourcecode-Management (CVS, SNV, etc.)
- | Report-Konfiguration (PMD, JUnit, Cobertura, etc.)
- | Profile
- | Properties

**x| Ergebnis** (Produkt) eines Projekts

Artifact-Typen:

- jar, war, ear, sar
- plugin, archetype, pom

**x| Artifacts** werden in Repositories verwaltet

**x| Jedes Projekt** erzeugt immer genau ein Artifact

**x| Es können eigene Artifact-Typen** definiert werden



### x| Goal

- | Kleinste Prozesseinheit  
„Unit Of Work“
- | Vergleichbar mit ANT  
Target



```
C:\WINDOWS\system32\cmd.exe
C:\>mvn compiler:compile_
```

### x| Plugin

- | Logische Gruppierung von  
Goals
- | Werden in den  
Repositories verwaltet

Syntax: **plugin:goal**

**x|** Zentrale Stelle, an welcher alle Artifacts abgelegt werden

- | Java Bibliotheken (jar, war, ear)
- | Projekt-Artifact (Releases, Snapshots)
- | Plugins
- | Archetypes (Projektvorlagen)

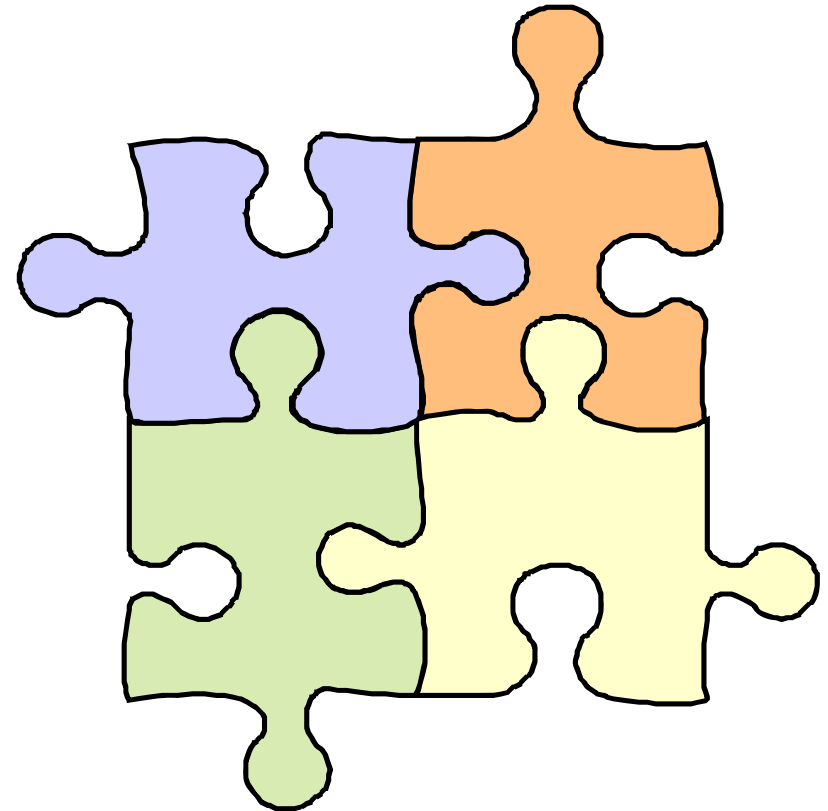
**x|** Maven 2 arbeitet üblicherweise mit zwei Typen von Repositories

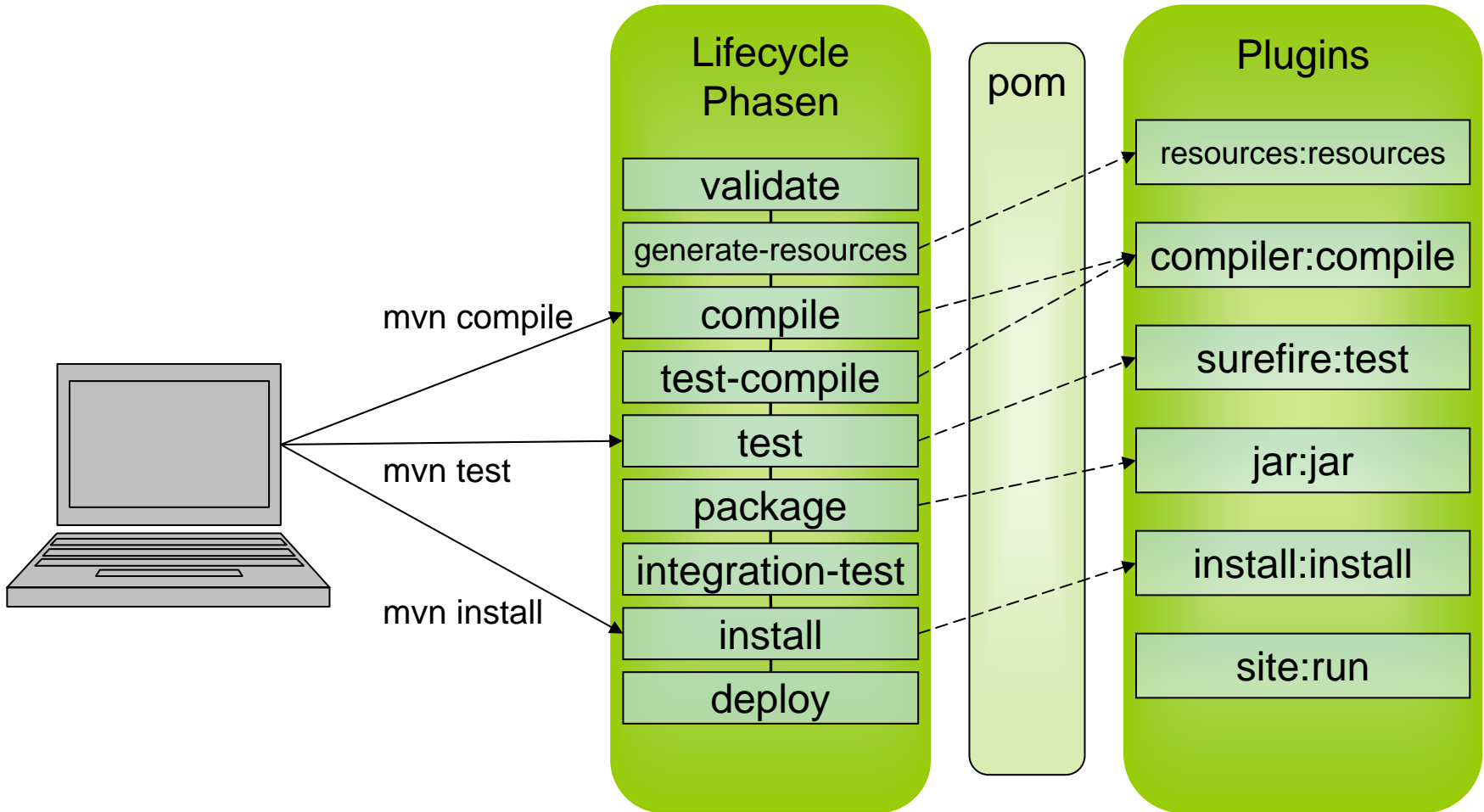
- | Lokales Repository                    **file://[user.home]/.m2/repository**
- | Remote Repositories                **http://repo1.maven.org**

Löst die Projektabhängigkeiten zu referenzierten Artifacts auf

- x|** Bibliotheken werden nicht physisch in das Projekt kopiert
- x|** Alle abhängigen Artifacts werden über die Repositories verwaltet

→ Kein „JARmageddon“



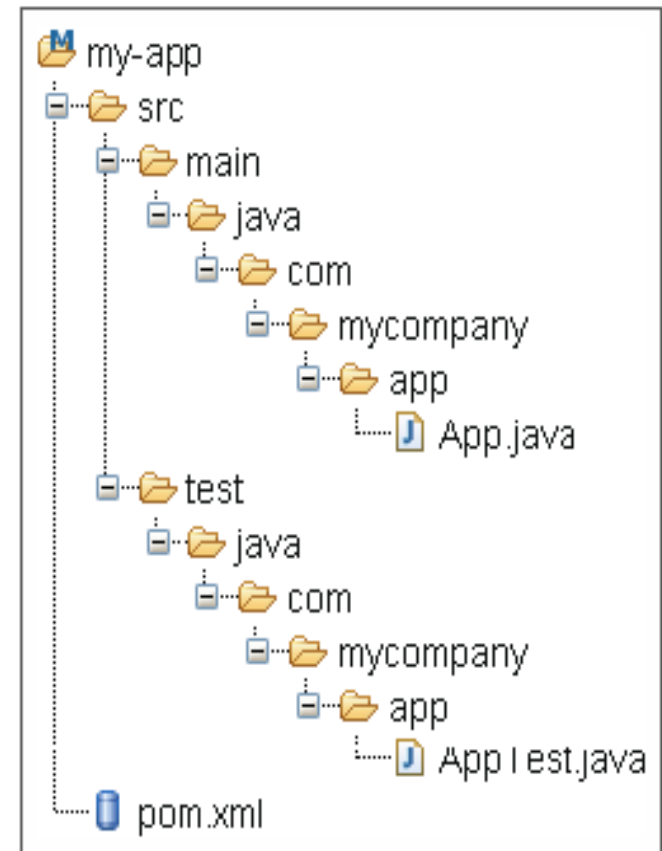


- x|** Jedes Projekt (POM) kann nur ein Ergebnis (Artifact) erzeugen
- x|** Definierte Projektstruktur für Quellen, Tests und Ressourcen
- x|** Global eindeutige Namen
  - | groupId
  - | artifactID
  - | Version

de.excellent:example1:1.0.0

→ **Geringe Einarbeitungszeit**

→ **Artifacts sind eindeutig identifizierbar**



The screenshot shows a Mozilla Firefox browser window with the address bar containing the URL: <https://www.excellent.de/cb/webfolder/maven2/site/mojo/plexxgen-maven-plugin/project-summary.html>. The page title is "plexxgen-maven-plugin Maven Mojo" and it was last published on 03/02/2007. The page content is organized into several sections:

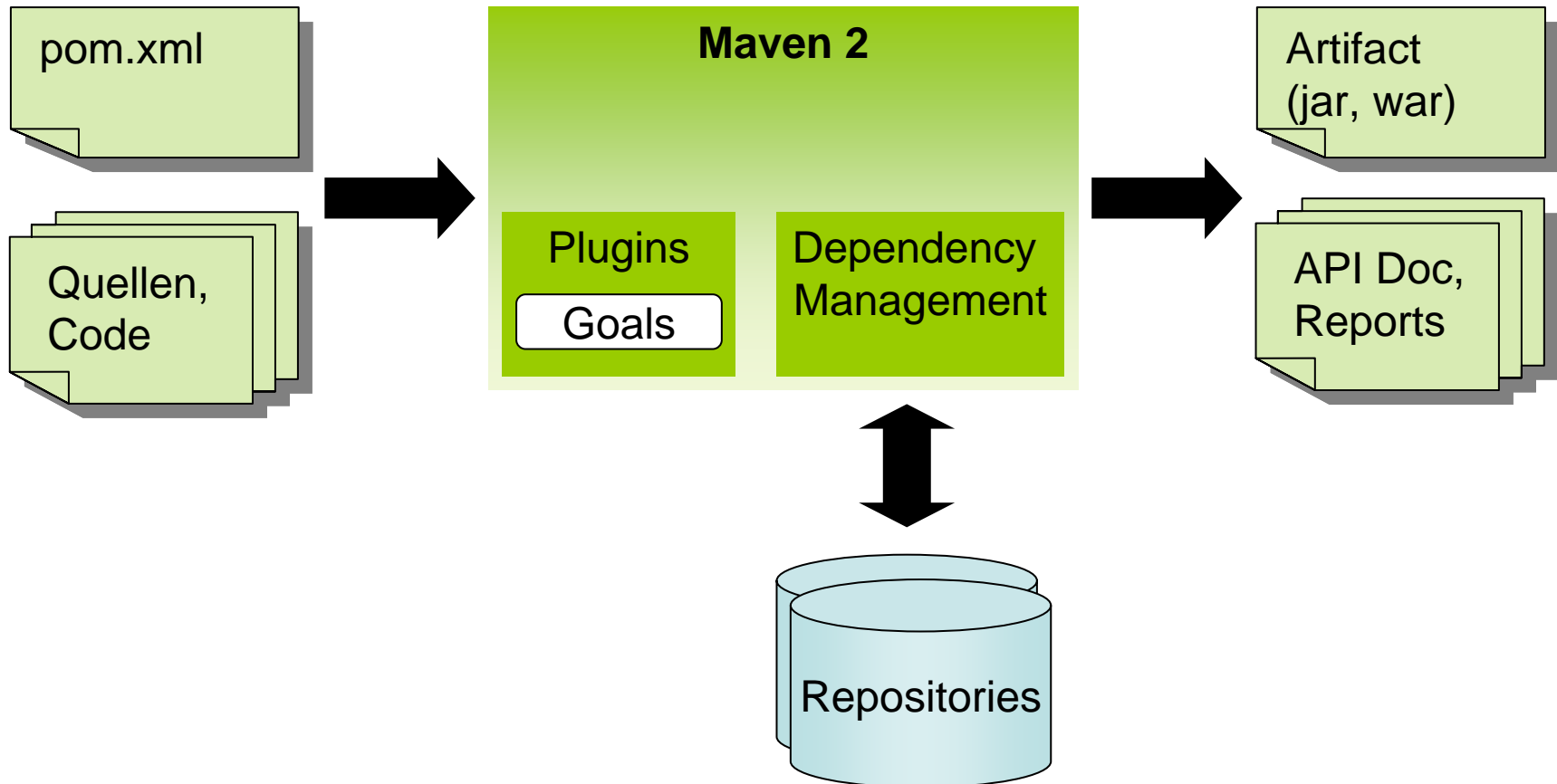
- Project Documentation:** A sidebar menu with links to Project Information, About, Continuous Integration, Dependencies, Dependency, Convergence, Issue Tracking, Mailing Lists, Project License, Project Summary (selected), Project Team, Source Repository, and Project Reports. Below the menu is a "built by maven" logo.
- Project Summary:** The main heading of the page.
- Project Information:** A table with the following data:
 

Field	Value
Name	plexxgen-maven-plugin Maven Mojo
Description	Plexx-Generator MOJO. Das MOJO erlaubt die Verwendung und die Konfiguration als Maven-Plugin.
Homepage	<a href="https://www.excellent.de/cb/proj/summary.do?proj_id=12/mojo/plexxgen-maven-plugin">https://www.excellent.de/cb/proj/summary.do?proj_id=12/mojo/plexxgen-maven-plugin</a>
- Project Organization:** A table with the following data:
 

Field	Value
Name	eXXcellent
URL	<a href="http://www.eXXcellent.de">http://www.eXXcellent.de</a>
- Build Information:** A table with the following data:
 

Field	Value
GroupId	de.excellent.maven.mojo
ArtifactId	plexxgen-maven-plugin
Version	0.0.10SNAPSHOT-SNAPSHOT
Type	maven-plugin

The footer of the page contains the copyright notice: © 2007 eXXcellent.



### x| Refactoring des bestehenden ANT Projektes

- | Projektstrukturen vereinfachen!
- | Bei mehreren Artifacts in Unterprojekte aufteilen

### x| Auflösen der Abhängigkeiten

- | Verwendete Bibliotheken identifizieren (Version)
- | Einspielen von 3rd-Party- und Inhouse-Bibliotheken in internes Repository

### x| POM definieren

- | Verzeichnisse konfigurieren
- | Bibliotheken einbinden
- | Tests und Reports Konfigurieren

### x| Spezial-ANT-Targets in Maven 2 umsetzen

- | Plugin-Konfiguration des Projekts erweitern
- | Eigene Plugins schreiben

### x| Weitere Features nutzen, um Effektivität und Qualität zu steigern

- x| Plugins nicht immer zu 100% dokumentiert
- x| 10% aller Projekte müssen individuell realisiert werden
- x| Eclipse-Plugin-Projekte werden derzeit noch nicht unterstützt
- x| Maven 2 funktioniert nur, wenn die Akzeptanz stimmt und die Generalisierung gelebt wird!

- x| Mehr Transparenz und bessere Verständlichkeit durch Projekt Standards, Konventionen, Richtlinien und Best Practices
- x| Wiederverwendbare Artifacts (Plugins / Bibliotheken)
- x| Einfach erweiterbar (Plugin-API)
- x| Tool – Integration (eclipse, idea, netbeans)
- x| 90% aller Projekte lassen sich über Standard Plugins realisieren
- x| Zentrales Repository für Abhängigkeiten
- x| Unterstützt qualitative Projektinformationen
- x| Aktive und passive Werkzeuge zur Qualitätssicherung

**→ Maven 2 hilft Zeit zu sparen und die Qualität zu verbessern!**

- x| Homepage**  
<http://maven.apache.org>
- x| Maven 2 Referenzkarte**  
<http://maven.apache.org/guides/MavenQuickReferenceCard.pdf>
- x| Englischs Tutorial**  
<http://cvs.peopleware.be/training/maven/maven2>
- x| Freies Online Buch**  
[http://www.mergere.com/m2book\\_download](http://www.mergere.com/m2book_download)
- x| Maven 2 POM Demystified**  
<http://www.javaworld.com/javaworld/jw-05-2006/jw-0529-maven>
- x| Maven 2 in der Praxis**  
[http://javamagazin.de/itr/online\\_artikel/psecom,id,767,nodeid,11](http://javamagazin.de/itr/online_artikel/psecom,id,767,nodeid,11)

**x| Diskussion / Fragen**

**x| Weiterführende Informationen gerne am Stand**  
| Ausstellungsbereich  
| Oliver Stauß ([o.stauss@exxcellent.de](mailto:o.stauss@exxcellent.de))

**x| Besten Dank!**