Maven

Table of contents

1. Apache Maven - Konfigurationsmanagement

Wenn man Maven unter LEO nachguckt, findet man die Übersetzung "Experte, Kenner".

In der Tat ist es so, dass das Buildmanagement-Tool Maven ein Expertentool ist. Mit keinem anderen Tool hat man über das Projekt einen so schnellen Eindruck als Entwickler, Contributer, Commiter, Project Leader oder welche Rolle man sich auch immer angeeignet hat.

Als ich vom Tool Maven im Jahre 2004 das erste Mal laß, wurde gerade Struts *mavenisiert* . So bezeichne ich den Vorgang, wenn aus einem ehemaligen (meist) ANT - Projekt ein Maven - Projekt wird. Das englische Pendant zu diesem Begriff würde sich hervorragend als Musiktitel für die chemischen Brüder eignen: Mavenize!

Diese Mavenisierung führte dann prompt dazu, dass man Struts in der aktuellen Version bauen konnte, ohne sich mühsam die entsprechenden Jars selbst aus dem Netz zusammenzuklauben und zu rauben.

Das war ein richtiges Heureka -Erlebnis und wäre ich in der Wanne gelegen...

Seitdem benutze ich, wann immer es geht, Maven und mavenisierte schon zahlreiche Projekte. Die größten Vorteile gegenüber Ant , dessen Marktdurchdringung meiner Meinung nach Maven überhaupt möglich machte, sind:

- Dependency Management
- Standardisierte Goals
- Project site Generierung und Reporting
- Build- und Project management und Project comprehension Tool

Maven ist nicht perfekt. Das wird schon daran deutlich, dass kurz nach Beginn der Entwicklung der Maven 1.* Version parallel die Maven 2 Version entstand.

Der zentrale Gedanke, der die Entwicklung von Maven 2 vorantrieb, ist *Convention over Configuration* oder zu deutsch *Konvention vor Konfiguration* . Der Spruch bedeutet, dass man als Entwickler, Architekt oder Projektleiter sich besser an vorgegebene Vereinbarungen hält, als alle Möglichkeiten der Konfigurierbarkeit auszuschöpfen und damit die Individualität eines Projekt zu fördern.

Eigentlich klingt das doch überaus positiv: *Projektindividualität* . Aber genau diese Eigenschaft macht es erforderlich, dass Entwickler, Architekten oder Projektleiter das Projekt mit all seinen Macken annehmen müssen und zur Vermeidung dieser Macken Workarounds bauen, die evtl. auch wieder individuell angelegt sind. Und so entsteht eine Reihe von Tools, Plugins, Workarounds und Regeln um ein Projekt, die man allesamt durch geeignete

Konventionen vermeiden hätte können. Dabei hilft einem Maven.