



ulm university

universität
uulm

Universität Ulm

Fakultät für Mathematik und Wirtschaftswissenschaften

Institut für angewandte Informationsverarbeitung

Begleit-Seminar zur Vorlesung

Entwicklung und Betrieb von Informationssystemen

Sommersemester 2014

Titel der Ausarbeitung:

Dokumenten- /Konfigurationsmanagement in verteilten

Software-Projekten

Sara Steisslinger	Matrikelnummer: 823825	Studienrichtung: WiWi BSc
Jasmin Klose	Matrikelnummer: 757415	Studienrichtung: WiWi BSc
Manuel Buchert	Matrikelnummer: 758373	Studienrichtung: WiWi BSc
Florian Rotter	Matrikelnummer: 761110	Studienrichtung: WiWi BSc
Daniel Glunz	Matrikelnummer: 702033	Studienrichtung: WiWi MSc
Manuel Ott	Matrikelnummer: 770969	Studienrichtung: WiWi BSc

Ehrenwörtliche Erklärung

Wir versichern, dass wir die Seminararbeit gemeinsam verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und uns auch sonst keiner unerlaubten Hilfe bedient haben.

Ulm, den _____

Unterschriften

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Gliederung	1
2	Software Configuration Management (SCM)	2
3	Konfigurationsidentifikation	3
4	Build Management/ System Building	4
4.0.1	Die Kosten eines fehlgeschlagenen Builds	5
5	Release Management	6
6	Versions-Management	7
7	Change Management	8
8	Fazit	9
	Abkürzungsverzeichnis	10
	Abbildungsverzeichnis	11
	Literaturverzeichnis	12

1 Einleitung

1.1 Motivation

1.2 Gliederung

2 Software Configuration Management (SCM)

Steuert und verwaltet verteilte Software-Projekte. Ganz normaler Text.

3 Konfigurationsidentifikation

4 Build Management/ System Building

Hat man nun den Quellcode der einzelnen Software Komponenten in dem Software-Projekt erstellt, dann möchte man diesen ausführbar machen. Hierum kümmert sich der Build Prozess. Dieser Prozess ist zuständig für das Kompilieren und Linking der einzelnen Software Komponenten in ein ausführbares System. Da die Abhängigkeiten von den Quellcodes, Bildern und weiteren Dateien bei großen Projekten sehr komplex und unübersichtlich sind, empfiehlt es sich, diese Abhängigkeiten nicht alle manuell zu kompilieren. Hinzu kann natürlich noch kommen, dass für unterschiedliche Systeme unterschiedliche Kombinationsmöglichkeiten von Einzelkomponenten für den Build Prozess benötigt werden. Man erkennt also, dass das einfache kompilieren von Quellcode für ein Software Projekt nicht ausreicht, sondern man weitreichende Überlegungen dazu anstellen muss.

Für diesen Build Prozessschritt gibt es inzwischen Unterstützung von automatischen Tools. Das bekannteste Tool ist make für C/C++. Weitere bekannte Tools sind Ant, Jam, Visual Studio und scons. Doch auch an solche automatischen Tools gibt es eine Anforderungsliste.

- Enthalten die Build Anweisungen alle benötigten Komponenten ?
- Ist für jede Einzelkomponenten die richtige Version spezifiziert?
- Sind alle benötigten Dateien verfügbar ?
- Sind die Referenzen innerhalb der Komponenten richtig, also ruft die eine Komponente eine andere mit den richtigen Parametern, Namen auf?
- Wird die Software auch für das richtige Betriebssystem erstellt?
- Mit welcher Kompilerversion und weiteren benötigten Software-Tools wird dieser Build- Prozess durchgeführt?

Prinzipiell gibt es zwei Aspekte für das Build Management. Zum einen ist das die Spezifikation, also das Erstellen des Build, und zum anderen die Ausführung von einem Build, also die Produktionsphase des Build Prozesses. Solch ein Build Prozess kann je nach Komplexität der zu erstellenden Software sehr rechenintensiv und lange sein.

4.0.1 Die Kosten eines fehlgeschlagenen Builds

5 Release Management

6 Versions-Management

7 Change Management

8 Fazit

Lorem ipsum dolet.

Abkürzungsverzeichnis

SCM Software Configuration Management

Abbildungsverzeichnis

Literaturverzeichnis