

Title

Improving Build Change Extraction with fine-grained hierarchical changes

Contact/Questions

Univ.- Ass. DI Christian Macho (christian.macho@aau.at), E.2.64

Context/Introduction

Build systems are widely used in today's software projects to automate integration and build processes. Similar to source code, build configuration need to be maintained to avoid outdated configurations, and build breakage as a consequence. Recent work indicates that neglected build maintenance is one of the most frequently occurring reasons why builds break. The context of this topic is situated in finding methods and developing tools that support the developers and release engineers in improving the quality of their build configuration.

Problem Description and Target

Prior studies have shown that classifying the changes that are made between two commits can help in studying the evolution and hence, investigating the impact of certain changes on, for example, quality. Recently, a tool named "BuildDiff" that is able to extract changes from Maven build configuration files has been developed and used to study frequent changes. It has been found useful to understand the evolution of build configuration files, for example when the most build changes were performed. The goal of this topic is to improve the existing classification with a set of finer change types (e.g. going deeper into the build specification) and to add the hierarchical dependencies that may exists in certain change types.

Tasks

- Improving the state-of-the-art tool in build change extraction
- Identifying fine-grained changes in build configuration files
- Automatically extracting fine-grained changes and their hierarchical coupling
- Evaluating the impact of the fine-grained changes on the evolution studies

Type

Praktikum/B.Sc. Thesis/ M.Sc. Thesis

Prerequisites

- Knowledge in Programming (preferably in Java)
- Knowledge in Software Engineering
- Basic Knowledge in Build Systems (i.e. Apache Maven)

Titel

Improving Build Change Extraction with fine-grained hierarchical changes

Kontakt/Fragen

Univ.- Ass. DI Christian Macho (christian.macho@aau.at), E.2.64

Kontext/Einführung

Build Systeme werden in heutigen Software Engineering Projekten immer häufiger eingesetzt, um die Integration und den Build Prozess zu automatisieren. Genauso wie Source Code muss auch die Konfiguration der Build Systeme gewartet werden, um zu verhindern, dass die Konfiguration veraltet und im schlimmsten Fall der Build Prozess fehlschlägt. Kürzliche Arbeiten zeigen, dass vernachlässigte Wartung der Build Konfiguration eine der häufigsten Ursachen für fehlgeschlagene Builds sind. Der Kontext dieses Themas liegt darin, Methoden zu finden und Tools zu entwickeln die die Entwickler und Release Engineers unterstützen, die Qualität der Build Konfiguration zu steigern.

Problem Beschreibung und Ziel

Einige Studien haben gezeigt, dass die Klassifizierung von Änderungen zwischen zwei direkt aufeinander folgenden Commits Studien zur Evolution von Änderungen unterstützt haben. Dadurch wurde es möglich andere Aspekte, wie zum Beispiel die Auswirkung von Änderungen auf die Qualität, zu untersuchen. Kürzlich wurde ein Tool namens „BuildDiff“ entwickelt, das Änderungen aus Maven Build Konfigurationen extrahieren kann. Es wurde verwendet, um die Häufigkeit von Änderungen in Build Konfigurationen zu untersuchen. Das Tool wurde ebenfalls verwendet, um eine erste Studie hinsichtlich der Evolution von Build Konfigurations Files und deren Änderungen zu untersuchen, zum Beispiel, um herauszufinden, wann im Laufe eines Projektes besonders viele oder besonders wenige Änderungen in der Build Konfiguration vorgenommen werden. Das Ziel dieses Themas ist es, das bestehende Tool um weitere feinere Änderungstypen zu erweitern und eine hierarchische Komponente aus den Änderungen zu extrahieren.

Aufgaben

- Weiterentwicklung des state-of-the-art Tools im Bereich Build Change Extraction
- Feine Änderungstypen in Build Konfigurations Files identifizieren
- Automatisiert Änderungen und deren Hierarchie extrahieren
- Auswertung der Auswirkung von Änderungen auf Studien zur Evolution von Projekten

Typ

Praktikum/B.Sc. Thesis/ M.Sc. Thesis

Voraussetzungen

- Programmierkenntnisse (vorzugsweise in Java)
- Software Engineering Kenntnisse
- Grundkenntnisse mit Build Systemen (Apache Maven)