Change Impact Analysis by Architecture-based Assessment and Planning

Kiana Rostami[1] Johannes Stammel[2] Robert Heinrich[1] Ralf Reussner[1]

Abstract: Software architecture presents the main artifact of software systems reflecting design decisions and thus influence their quality attributes. Furthermore, during software evolution each architecture decision also influences technical artifacts (e.g., test cases) and the corresponding organizational responsibilities (e.g., tester). Thus, it is important to predict the impact of a change request (e.g., changing an interface) on the software architecture and other software artifacts for decision-making. Hence, a software architect can estimate the effort of the implementation of a change request due to corresponding implementation tasks. However, existing approaches are limited to artifacts of the software development process or do not use formal architecture descriptions. We present the Karlsruhe Architectural Maintainability Prediction (KAMP), that enables software architects to analyze the propagation of change requests in software architecture models. Our approach is not limited to the technical operations but as well considers the organizational tasks. KAMP supports software architects by automatically generated task lists to implement changes. In an empirical study, we showed, that KAMP improves the scalability, precision, and completeness of change propagation analysis.

This is an extended abstract of the paper *Architecture-based Assessment and Planning of Change Requests* published in *QoSA'15* proceedings [Ro15].

Keywords: Software Evolution, Change Request, Impact Analysis

1 Introduction

Software architecture is the central artifact of a software system. A software architecture model helps project members in decision making processes such as reuse or effort estimation. During its life cycle the software architecture is subject to change due to changes in its environment. As several project members in various organizational units are involved during software development and maintenance, it is important to predict the impact of change requests to coordinate the implementation tasks. Further, implementing a change request involves changing all software artifacts such as code, test cases, build specification, configuration files or documentation. On the other hand, a change request can be implemented in various ways resulting in different impact on the software artifacts and different quality attributes of the resulting software architecture. Thus, predicting impact of a change request in a software system can help project members to better predict and plan the maintenance effort during software development and maintenance.

[1] Karlsruhe Institute of Technology, {rostami, heinrich, reussner}@kit.edu
[2] andrena objects ag, johannes.stammel@andrena.de

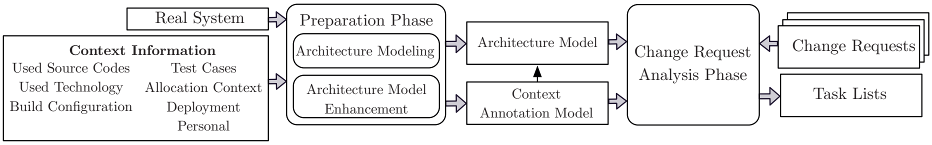

Fig. 1: Overview of the approach [Ro15]

2 Karlsruhe Architectural Maintainability Prediction (KAMP)

KAMP is a scenario-based approach [Ro15], that analyses the change propagation in the software system model based on an initial change request. Using KAMP project members can annotate the software architecture model with context information such as test cases or deployment information. Annotating software architecture models allow to derive a task list involving organizational and technical tasks. Hence, project members can estimate the effort of the implementation of a change request due to corresponding implementation tasks. Fig 1 illustrates the KAMP approach. It comprises two phases: i) In the preparation phase project members model the software architecture using Palladio Component Model (PCM) [Re16] and annotate it with context information. ii) In the change request analysis phase they select the initial changed elements in the architecture model. Then, KAMP can automatically calculate the structural change propagation in the model of the software system using a set of predefined rules (e.g., change propagation from data types to interfaces). The output of KAMP is a task list involving potentially changed software artifacts.

The main contribution of KAMP is the calculation of a task list required to implement change requests. We conducted an empirical study with a treatment group, a control group, and an expert group to validate the quality of the task lists. The results of the empirical study showed, that KAMP improves the scalability of the change propagation analysis and the automatically generated task list are more precise and complete than manually ones.

As future work we will apply KAMP for change impact analysis in other domains such as manufacturing automation systems [Vo15] and for supporting adaptation planning [He16].

Literatur

[He16] Heinrich, Robert: Architectural Run-time Models for Performance and Privacy Analysis in Dynamic Cloud Applications. SIGMETRICS Perform. Eval. Rev., 43(4):13–22, 2016.

[Re16] Reussner, Ralf et al.: Modeling and Simulating Software Architectures – The Palladio Approach. MIT Press, 2016.

[Ro15] Rostami, Kiana et al.: Architecture-based Assessment and Planning of Change Requests. In: Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures. QoSA ’15. ACM, S. 21–30, 2015.

[Vo15] Vogel-Heuser, Birgit et al.: Architecture-based Assessment and Planning of Software Changes in Information and Automated Production Systems. In: IEEE International Conference on Systems, Man, and Cybernetics. 2015.