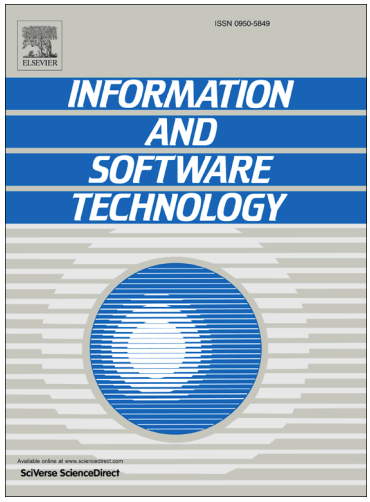

Validating a Model-Driven Software Architecture Evaluation and Improvement Method: A Family of Experiments

Javier Gonzalez-Huerta, Emilio Insfran, Silvia Abrahão, Giuseppe Scanniello

PII: S0950-5849(14)00135-9
DOI: http://dx.doi.org/10.1016/j.infsof.2014.05.018
Reference: INFSOF 5475

To appear in: *Information and Software Technology*

Received Date: 17 October 2013
Revised Date: 22 May 2014
Accepted Date: 23 May 2014

Please cite this article as: J. Gonzalez-Huerta, E. Insfran, S. Abrahão, G. Scanniello, Validating a Model-Driven Software Architecture Evaluation and Improvement Method: A Family of Experiments, *Information and Software Technology* (2014), doi: http://dx.doi.org/10.1016/j.infsof.2014.05.018

Validating a Model-Driven Software Architecture Evaluation and Improvement Method: a Family of Experiments

Javier Gonzalez-Huerta[1], Emilio Insfran[1], Silvia Abrahão[1], Giuseppe Scanniello[2]

[1] Department of Information Systems and Computation, Universitat Politècnica de València, Valencia, Spain
[2] DiMIE, Università della Basilicata, Potenza, Italy

Abstract:

Context: Software architectures should be evaluated during the early stages of software development in order to verify whether the Non-Functional Requirements (NFRs) of the product can be fulfilled. This activity is even more crucial in Software Product Line (SPL) development, since it is also necessary to identify whether the NFRs of a particular product can be achieved by exercising the variation mechanisms provided by the product line architecture or whether additional transformations are required. These issues have motivated us to propose QuaDAI, a method for the derivation, evaluation and improvement of software architectures in model-driven SPL development.

Objective: We present in this paper the results of a family of four experiments carried out to empirically validate the evaluation and improvement strategy of QuaDAI.

Method: The family of experiments was carried out by 92 participants: Computer Science Master's and undergraduate students from Spain and Italy. The goal was to compare the effectiveness, efficiency, perceived ease of use, perceived usefulness and intention to use with regard to participants using the evaluation and improvement strategy of QuaDAI as opposed to the Architecture Tradeoff Analysis Method (ATAM).

Results: The main result was that the participants produced their best results when applying QuaDAI, signifying that the participants obtained architectures with better values for the NFRs faster, and that they found the method easier to use, more useful and more likely to be used. The results of the meta-analysis carried out to aggregate the results obtained in the individual experiments also confirmed these results.

Conclusions: The results support the hypothesis that QuaDAI would achieve better results than ATAM in the experiments and that QuaDAI can be considered as a promising approach with which to perform architectural evaluations that occur after the product architecture derivation in model-driven SPL development processes when carried out by novice software evaluators.

Keywords: Software Architectures, Software Architecture Evaluation Methods, Quality Attributes, ATAM, Family of Experiments, Meta-analysis.

1. Introduction

Software architectures are a means to preclude or permit the achievement of the Non-Functional Requirements (NFRs) of a software system. In Software Product Line (SPL) development, in which a set of software intensive systems sharing a common managed set of features are developed from a common set of core assets, the product line architecture should contain variation mechanisms that help to achieve a set of permitted variations, including functional, structural and quality concerns [23]. The product architecture is derived from the product line architecture by exercising its built-in architectural variation mechanisms, which support both the functional and NFRs for a specific product.

Once it has been derived, the product architecture should be evaluated to assess the achievement of the product's specific requirements. When the required levels of quality attributes for a specific product fall outside the original specification of the SPL (and cannot be attained by using product line variation mechanisms), certain architectural transformations should be applied to the product architecture to ensure that these NFRs are met [15].