

# Preface

## *Scope*

Two essential activities in the software development process are requirements engineering (RE) and software architecting. The focus of RE is on eliciting, analyzing, and managing requirements. Software architecting is concerned with providing an abstraction of the system as a blueprint to manage the complexity of software systems. The development of software architectures is a challenging task, even when the requirements for a software system are clear. Requirements in general and quality requirements in particular drive the architecture of a software system, whereas decisions made in the architectural phase can affect the achievement of initial requirements and thus change them. The common way of traditional software development processes such as the waterfall model is to build a software architecture from requirement descriptions. This process considers the forward development process from requirements to the software architecture, it however does not consider the impact of design decisions on initial requirements. The problem of the linear software development processes is twofold. On the one hand requirements are elicited, analyzed, and specified in isolation without considering the impact of architecture artifacts. On the other hand, design decisions are made without managing the conflicts and making necessary changes in the requirements. Hence, requirements and software architecture evolve together. According to the intertwining nature of requirements and architectures at each level of refinement, requirement descriptions cannot be considered in isolation and should be co-developed with architectural descriptions iteratively and concurrently. There is, however, no structured solution on how to perform the co-development of requirements and software architecture. With this book, we aim at providing a com-

prehensive and structured approach that supports the intertwining relationship of requirements and software architecture. We propose a framework for the *Problem-oriented and Quality-based Co-Development of Requirements and Architecture* (QuaDRA). QuaDRA guides the software engineer in co-developing the requirements and early software architecture in an iterative and concurrent manner, taking into account quality requirements.

## ***Content***

This book first systematically identifies the lack of methodological support for development of requirements and software architecture in the state-of-the-art. We systematically derive the meta-requirements for such a method. Later we use the extracted meta-requirements as evaluation criteria for building a comparative evaluation framework aiming at analyzing the state-of-the-art methods. To gather the state-of-the-art, we conducted a systematic literature review. Applying the evaluation framework to the state-of-the-art showed that none of the compared methods fulfills all the meta-requirements. To close this gap, this book proposes the QuaDRA framework as a problem-oriented approach comprising eight phases. It provides an instantiation of the twin peaks model, in which we move back and forth between two peaks for co-developing requirements and software architecture. QuaDRA includes several structured methods. These methods guide software engineers in quality- and pattern-based co-development of requirements and early software architecture design alternatives in an iterative and concurrent manner. The QuaDRA framework provides support for developing a single system. We further show how to enhance it for supporting a software product line development. Finally, we validate the QuaDRA framework by applying the systematic evaluation framework. The evaluation framework provides a basis for comparing QuaDRA with the state-of-the-art based on the previously defined evaluation criteria. The comparative evaluation demonstrates that QuaDRA exhibits a substantial progress over the state-of-the-art.

## ***Audience***

This book is aimed at practitioners such as software engineers working in the areas of requirements engineering and software architecture design. Particularly, novices and less experienced software engineers benefit from this book as it pro-

vides detailed guidance on how to develop software architectures from quality requirements in a systematic way. It is also intended for researchers who aim at investigating the relationship between requirements engineering and software architecture and how the activities in each phase restrict the scope of consideration in the other phase. As this book provides a novel software development method compared to the traditional software development approaches, it can be served as a supplementary reading for the undergraduate courses in the software engineering discipline. This book pays a particular attention to the quality requirements security and performance. Therefore, it can be used to teach the graduate courses in the requirements engineering and software architecture with focus on quality requirements and their systematic integration into the software architecture.

Cologne Germany,  
November 2016

*Azadeh Alebrahim*

<http://www.springer.com/978-3-658-17693-8>

Bridging the Gap between Requirements Engineering  
and Software Architecture

A Problem-Oriented and Quality-Driven Method

Alebrahim, A.

2017, XXVI, 500 p. 141 illus., Softcover

ISBN: 978-3-658-17693-8