

A Survey of Software Architecture Decision-Making Techniques

Lihua Xu, Debra J. Richardson, and Hadar Ziv
Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3425
{lihuax, djr, ziv} @ics.uci.edu

ISR Technical Report # UCI-ISR-07-10

December 2007

Abstract: Software quality attributes describe both the specific criteria related to how the system is built (e.g. cost, development time), and qualitative constraints on various attributes of functions or services that the system should provide (e.g. performance, usability, reliability). These requirements concern not only the customers for whom the system is produced but ultimately every stakeholder involved with the software. Unlike functional requirements relating to the common services a system should provide, and on which stakeholders must agree, quality requirements usually differs from system to system, from stakeholder to stakeholder. These quality attributes can only be “satisficed” [CNYM00a], rather than “accomplished” or “satisfied”, since design decisions can contribute only partially towards or against a particular quality attribute of the system. Moreover, these quality attributes are often inter-connected, whether in agreement or in conflict, with each other. Hence, reaching an agreed understanding of these qualities attributes, and finding the optimal balance among them instead of studying a single one in isolation, are crucial in achieving a high quality software product.

Software architecture sets the boundary of systematic reasoning about various quality attributes that are relevant to the system domain. A high-quality software architecture facilitates the development of a high quality software system. Designing such architecture usually involves a set of interdependent design decisions that contribute to quality attributes differently; the architect must not only iteratively explore different design alternatives for each design decision, but also consider the interplay among them and balance the myriad tradeoffs from conflicting quality attributes. This explorative process is an incremental decision making process in which the architect evaluates the design alternatives with respect to the quality attributes, and reaches an optimized design that fulfills stakeholders’ requirements.

To inform these design decisions, software engineers propose architecture analysis techniques to analyze each design alternative, compare them, and understand their differences. This survey studies existing architecture analysis approaches that address all required quality attributes of the system, from the perspective of how they support an explorative design process with regard to the quality attributes. In particular, the survey explores the approaches from five perspectives: the support for gathering requirements from multiple stakeholders and resolve conflicts; the support for modeling architecture and quality attributes and identify design decisions involved in the architecture; the support for analyzing and comparing design alternatives for each design decision; the support for considering all design decisions and their interdependencies; and the automated support for the process.

¹ Quality requirements and quality attributes are interchangeable in this paper.