Good software architecture is one of the key factors in successfully developing and evolving a system or a family of systems. Software architecture provides the key framework for the earliest design decisions taken to achieve functional and quality requirements. In addition, it has a profound influence on project organizations' functioning and structure. Poor architecture usually results in project inefficiencies, poor communication, and poor decision making. Software architecture for a family of systems also helps identify the commonality among different systems and explicitly document variability. Since software architecture plays a significant role in the life of a system, it is important to evaluate a system's architecture as early as possible. Architecture evaluation is considered one of the most important and effective techniques of addressing quality related issues at the software architecture level and mitigating architectural risks. Moreover, architecture evaluation sessions are an effective means of sharing and capturing architecture design rationale, reasoning behind architecture design decisions. This tutorial highlights the benefits and challenges in evaluating software architectures. It discusses theoretical and practical concepts underpinning some of the well-known scenario-based architecture evaluation methods and various approaches to characterize quality attributes using scenarios. The use of the presented methods, techniques, and tools will be demonstrated with a case study based on an industrial project.

Biography

M. Ali Babar is a senior researcher with Lero, the Irish Software Engineering Research Centre, where he leads projects on software architectures and technology evaluation in the software product line research area. Previously, he was working as a researcher with National ICT Australia (NICTA), where he initiated and led several research and development projects in the areas of software architecture, architectural knowledge management, and process improvement using empirical methods. During his stay at NICTA, he designed and developed various methods and tools in the area of software architecture knowledge management, which attracted significant industrial collaboration and funding. He has authored/co-authored more than 50 publications in peer-reviewed journals, conferences, and workshops. He has also presented tutorials in the area of software architecture knowledge management at various international conferences including ICSE 2007, SATURN 2007 and WICSA 2007. Prior to joining research and development, he worked as a software engineer and an IT consultant for several years in Australia. He received a Ph.D. degree in Software Engineering from the University of New South Wales, Australia. His current research interests include software product lines, software architecture design and evaluation, architecture knowledge management, tooling supporting, and empirical methods of technology evaluation.