ARCHITECTURE BASED SOFTWARE QUALITY MAINTENANCE
RIPPLE EFFECT ANALYSIS

Sajid Anwar¹, Awais Adnan¹, Masoom Alam¹, Tamleek Ali Tanveer¹
Muhammad Ali¹, Muhammad Ramzan², Arfan Jaffer³
Arshad Ali Shahid³ and Abdul Rauf³

¹Department of Computer Science
Institute of Management Sciences Peshawar
7/B3, Phase 5, Hayatabad, Peshawar
{sajid.anwar; awais.adnan; masoom.alam; tamleek; muhammad.ali}@imsciences.edu.pk

²University of Arid Agriculture
Shamsabad, Murree Road, Rawalpindi, Pakistan
ramzan@uaar.edu.pk

³Department of Computer Science
FAST National University of Computer and Emerging Sciences
A. K. Brohi Road, H-11/4, Islamabad, Pakistan
{arshad.ali; arfan.jaffer; a.rauf}@nu.edu.pk

Received July 2010; revised November 2010

ABSTRACT. One of the major challenges for software developer is to fulfill the quality requirements of the software systems. This emphasis on software quality has some serious implications in terms of customer satisfaction and system acceptance. Due to its significance, it is also considered as one of the major challenges to be met by software developer since s/he is responsible for fulfilling the quality requirements of the software systems. One way to address this challenge is to adopt architecture based software development. Software architecture as an artifact can be used to deal with software quality attributes (QA) such as maintainability, performance and reliability. Recently, the focus of business is transforming rapidly from manual to computer based automations. Resultantly, software intensive systems are becoming large and complex. This is enhancing our emphasis on system quality which ultimately gives rise to the need for software quality maintenance. In order to optimize the quality of the software maintenance in a rapidly changing environment, study of its ripple effect is very significant. In this paper, a methodology for architecture based quality maintenance ripple effect determination and analysis is proposed. A QA’s Property-component connectivity matrix and component connectivity matrix for ripple effect analysis is also described. The methodology is illustrated and evaluated by using web content extraction application architecture. The advantages of the proposed methodology are also presented.

Keywords: Software maintenance, Ripple effect analysis, Quality attributes

1. Introduction. The phenomenon of ripple effect in software development is not new. The term ripple effect was first introduced by Haney [1] using a technique called “Module connection analysis”. This technique was based on the idea that changes in one module necessitate a change in any other module/modules. Several other software maintenance models such as SADT [2] and Methodology for Software Maintenance [3] have considered ripple effect measure as a necessary factor. Software maintenance consumes enormous number of organization’s overall resources and has some serious implications in terms of cost and effort [4]. Typically, at least 60% of software life cycle cost is associated with software maintenance activities [5]. So, ripple effect analysis based maintenance technique