A Comparative Analysis of Reliability Assessment Methods for Web-based Software

Jinhee Park, Korea Advanced Institute of Science and Technology, Daejeon, South Korea
Yeong-Seok Seo, Korea Advanced Institute of Science and Technology, Daejeon, South Korea
Jongmoon Baik, Korea Advanced Institute of Science and Technology, Daejeon, South Korea

ABSTRACT

As web technology has advanced, many business software applications are built on the web. In such web environment, it has become very important to ensure the reliability of web-based software systems such as Software as a Service (SaaS) or Service-Oriented Architecture (SOA) based systems because service failures in those systems may have an effect on extensive users. With the comparison to the reliability studies on traditional software, there are only a few studies on the reliability of web-based software. The dynamic environment of the web makes it much more complicated to assess the reliability of web-based software. In this paper, the authors investigate the characteristics of reliability assessment methods for web-based software such as SaaS and SOA based software systems. The authors also evaluate those methods based on hypothetical execution scenarios to analyze the strengths and weaknesses of each method. This analysis helps the authors identify remaining problems on the reliability research in the web environment and provides insight into possible solutions.

Keywords: Reliability Assessment, Service-Oriented Architecture (SOA), Software as a Service (SaaS), Web Application, Web-Based Software

INTRODUCTION

With the shift of a software development paradigm, web-based software has gained in popularity. Since the web environment is easy to access and provides immediate feedback on software modifications, software in this environment is very attractive to the companies that attempt to reduce development costs and respond to rapid technological advance. In recent years, as the flexibility towards a platform and the reusability of software are emphasized, Service-Oriented Architecture (SOA) has been developed. SOA based systems are now providing various services to potential worldwide users, and the IT infrastructures of many companies have been commonly implemented as the SOA based systems. More recently, as a cloud computing service market has rapidly grown, the concept of IT assets is shifting from owning to renting. While most of companies individually owned software assets in the past,
they are widely renting software systems by adopting Software as a Service (SaaS) delivering software via the web. With the fast growth of the cloud computing market, its scale is expected to reach $420 billion in 2012 from $160 billion in 2008 (Mirzaei, 2008). According to the announcement of Gartner, 80% of the Fortune 1,000 companies would use some form of cloud services by 2012 (Gartner, 2008).

In spite of the popularity of web-based software systems, they still suffer from various critical failures. These failures or service interruptions often cause extensive damages because they affect massive users. For example, due to the disabled cloud center of Amazon, the cloud services such as Foursquare (location-based SNS), Quora (social Q&A site), Zynga (social gaming site) have been discontinued for about thirty minutes in April 2011 (Amazon, 2011). It badly affected many of the Fortune 1,000 companies and hundreds of others. In addition, the overloaded Gmail servers due to the changes of a re-routing code caused a global outage in September 2009 (Gunawi et al., 2012). Since such impacts on users (i.e., individuals or companies) can be fatal, it is necessary to assure the high quality of web-based software systems. In particular, software reliability, which is often used for estimating the current state of software quality and predicting the occurrence of failures, is a precondition for a software quality assurance.

To assess the reliability of traditional software, multiple types of reliability models have been developed over the 40 years (Lyu, 1996). In particular, Software Reliability Growth Models (SRGMs) for a monolithic application have been often used to verify that software has achieved the reliability objective before release. They provide an analytical framework for describing a phenomenon that the reliability grows as the residual faults of software are removed during operational testing. Test managers can determine a software release date by assessing the software reliability at certain times of testing. As another major class of software reliability models, Component Based Reliability Models (CBRM s) have been employed to predict the software system reliability. These models reflect the internal structure of software and are applicable in the software design phase (Goseva-Popstojanova & Trivedi, 2001). They are usually classified into the four groups such as additive models, path based models, state based models, and UML based models. Using these models, we can find the optimal architecture in terms of reliability and understand how the reliability of each component affects the reliability of an entire software system.

However, the focus of software reliability studies still stays on traditional software in spite of the shift of a software paradigm into the web environment. Unlike traditional software, web-based software has the unique characteristics such as unpredictable network condition, huge user population, cross-platform support, and data security requirements (Hsu & Huang, 2009). According to a report from IT companies such as Google and Microsoft (Pertet & Narasimhan, 2005), while reliability approaches for the traditional software concentrate on logical or computational errors of software systems, main causes of failures in the web environment are network errors, configuration errors, system overload, and resource exhaustion (Rahmani, Azadmanesh, & Siy, 2011). It means that, unlike the common belief that a software failure rate is constant without the modification of source code, the failure rate of web-based software may change according to network conditions or workload information. Due to these characteristics, we cannot simply apply the reliability models designed for the traditional software into the web-based software. Although some techniques exist to measure and assure the web-based software reliability, there are still many new challenges.

In this paper, we analyze the existing methods to assess the reliability of the web-based software. Reliability assessment methods for a web application, a SOA based system, and SaaS are compared. We use hypothetical scenario-based evaluation to identify the strengths and weaknesses of each method. The scenarios describe the dynamic web environment, and we analyze how well each reliability
12 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:
www.igi-global.com/article/a-comparative-analysis-of-reliability-assessment-methods-for-web-based-software/103280?camid=4v1

www.igi-global.com/e-resources/library-recommendation/?id=2

Related Content

Enhancing the Daily Routines of Equine Veterinarians using Mobile Technology: The m-Equine Case
www.igi-global.com/article/enhancing-the-daily-routines-of-equine-veterinarians-using-mobile-technology/89371?camid=4v1a

Combining Requirements Engineering and Agents
www.igi-global.com/chapter/combining-requirements-engineering-agents/28403?camid=4v1a
Multirate Systems and Filterbanks
[www.igi-global.com/chapter/multirate-systems-filterbanks/27223?camid=4v1a](www.igi-global.com/chapter/multirate-systems-filterbanks/27223?camid=4v1a)

Automated Context Formalization for Context-aware Specification Approach
[www.igi-global.com/article/automated-context-formalization-for-context-aware-specification-approach/218170?camid=4v1a](www.igi-global.com/article/automated-context-formalization-for-context-aware-specification-approach/218170?camid=4v1a)