Impact of Fault-Prone Components on Effective Software Testing: An Industrial Survey

D. Jeya Mala, Department of Computer Applications, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India
A. Jalila, Department of Computer Applications, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India

ABSTRACT

In a real-time software system, often a small portion of the components holds the major operations of the system. Thus, the rate of failure of those portions is high, which leads to major loss to the system. In order to develop cost-effective and high-quality software systems, those fault-prone components must be identified and tested rigorously. This paper analyzes the industrial questionnaire survey with a focus on the significance of early identification of fault-prone components within a system and importance of the fault-prone components-based testing. From the survey results, it has been inferred that, fault-prone components can be identified using software models, formal specifications and appropriate metrics and measures; which would, in turn, reduce the cost of defect and testing. Furthermore, the present study endeavours to investigate as to how far research efforts are essential in providing guidelines in the adoption of fault-prone components-based development, thus improving the quality of software development.

Keywords: Defect, Fault-Prone Components, Formal Specification, Metrics, Software Testing

INTRODUCTION

Software testing is one of the major quality criteria to guarantee the reliability of a system. In other words, high cost and development effort are involved in the process of testing of complex systems. As per the recent studies, it has been observed that the existing methods applied for testing do not satisfy all the testing requirements (Bertolino, 2007 & Farooq, 2013). This often leads to incomplete testing, which in turn, leads to compromises in terms of quality; therefore, software test effort reduction is an important research topic in recent years. However, the earlier research works have widely acknowledged that Fault-Prone Components-based (FPC) testing
is an effective approach to minimize testing effort (Frank, Alla, Jürgen & Robert, 2012; Ray, Lal & Mohapatra, 2011). Hence, these components must be rigorously analyzed and designed for effective testing.

According to Tomaszewski Piotr et al. (2006) FPC prediction from early software life cycle phases are more accurate than during later development phases. In order to evaluate the early identification of FPC in industrial strength applications, an industrial survey has been conducted, with a special focus on the stakeholder perspectives on some contemporary aspects related to the impact of FPC in software failures and fault-prone components-based testing. Based on this survey, a number of interesting observations have been derived. This provides an insight into the importance of rigorous testing of FPC prior to the design phase. This survey has also drawn conclusions that none of the testing tools available in the market as of now is able to find such fault-prone components; thus, development of a technique or a tool to address such problem is the need of the hour, so as to ensure quality. By means of qualitatively and quantitatively analyzing the results of the survey questionnaire, it is believed that the survey results can assist in providing the necessary directions for fault-prone components-based software development. The contributions of this paper are three-fold, as given below:

• Underline the impact of fault-prone components in cost-effective software development;
• Provide the insightful analysis on the measured for early identification of FPC from specification;
• Set a basis for the future research trends focus on the FPC-based software development

The remainder of this article is structured as follows. Section 3 starts with a detailed description of the related work. The method adopted for the survey has been presented in Section 3, while Section 4 analyzes the survey results. Conclusions arrived at as a result of the above study have been discussed in Section 5.

RELATED WORK

Table 1 clearly illustrates that significant work has been done for predicting fault-prone components at various stages of software development such as design level and source-code level. In Table 1, the second column presents the methodology adopted by various approaches and the third column shows name of the metrics applied. The fourth column provides the details of critical path extraction.

It is evident from the existing research works, as shown in Table 1, that design documents and source codes have been used as the major sources for the prediction of fault-prone components within a system. Most of the studies have used object-oriented design metrics, more specifically, CK metrics, for fault-prone components identification. Few earlier research studies have used statistical and machine learning approaches to classify components as fault prone and non-fault prone components. However, the outcome of the present survey throws light on a new trend towards the identification of fault-prone components, namely, using formal specifications.

SURVEY METHOD

The survey was performed based on a Questionnaire prepared by the authors, and the invitation was distributed among industrial software development companies. Approximately, 60 software professionals from different software organizations in major cities have participated in this survey.
The Role of Foundational Ontologies for Domain Ontology Engineering: An Industrial Case Study in the Domain of Oil and Gas Exploration and Production
[www.igi-global.com/article/role-foundational-ontologies-domain-ontology/43606?camid=4v1a](www.igi-global.com/article/role-foundational-ontologies-domain-ontology/43606?camid=4v1a)

A Survey of Development Methods for Semantic Web Service Systems
[www.igi-global.com/chapter/survey-development-methods-semantic-web/29400?camid=4v1a](www.igi-global.com/chapter/survey-development-methods-semantic-web/29400?camid=4v1a)

Control Architecture and Controllability
[www.igi-global.com/chapter/control-architecture-and-controllability/179499?camid=4v1a](www.igi-global.com/chapter/control-architecture-and-controllability/179499?camid=4v1a)