Chapter 97
Bug Handling in Service Sector Software

Anjali Goyal
JIIT, India

Neetu Sardana
JIIT, India

ABSTRACT

The technology enabled service industry is emerging as the most dynamic sectors in world's economy. Various service sector industries such as financial services, banking solutions, telecommunication, investment management, etc. completely rely on using large scale software for their smooth operations. Any malwares or bugs in these software is an issue of big concern and can have serious financial consequences. This chapter addresses the problem of bug handling in service sector software. Predictive analysis is a helpful technique for keeping software systems error free. Existing research in bug handling focus on various predictive analysis techniques such as data mining, machine learning, information retrieval, optimisation, etc. for bug resolving. This chapter provides a detailed analysis of bug handling in large service sector software. The main emphasis of this chapter is to discuss research involved in applying predictive analysis for bug handling. The chapter also presents some possible future research directions in bug resolving using mathematical optimisation techniques.

INTRODUCTION

"Prevention is better than cure" this proverb applies to the software industry significantly and since service sector industry is taking a software step nowadays, the proverb clearly applies to service sector as well. Bugs make the software system fail to meet the user requirements and make customers unhappy. Recent market studies have shown that software bugs may lead to serious financial concerns and can be dangerous for any organization. In early 2016, HSBC became the first bank to suffer a major IT outage. The bank blamed a complex technical issue for this incident. Another incident in August 2015, HSBC

failed to process 275,000 individual payments which left many without any pay before the bank holiday weekend. The reason was a bug in electronic payment system. In June 2015, due to a software failure 600,000 payments failed to enter the accounts of The Royal Bank of Scotland (RBS) overnight. Earlier in 2012, 6.5 million customers of RBS experienced an outage due to batch scheduling software which led to a fine worth £56 million for the bank (Jee, 2016). The implications from these situations highlight the need to consider bug handling in service sector software seriously. If a bug is diagnosed efficiently it prevents the high efforts and cost incurred in the rectification of defect. Thus the software bugs need to be dealt with highest priority.

Large software development projects usually use a bug tracking systems (BTS) for gathering and organizing bugs reported by the developers and users. The use of BTS improves the software product's quality as it allows more bugs to be identified and resolved (Raymond, 1998). But the high volume of submitted bug reports makes bug handling a cumbersome task. For instance, the popular open source project Eclipse received 3426 bug reports during four-month period (01, January 2005 to 30, April 2005), averaging 29 bugs being reported daily (Anvik et al., 2006). Thus "effective resolving process of bug reports in software systems" becomes a research hotspot. (Erlikh, 2000) stated that software maintenance and evolution activities constitute 90% cost of total software development. To reduce the cost and efforts in bug handling, large scale open source software projects employ teams that utilize BTS such as Bugzilla[1], JIRA[2], FogBugz[3], etc. to keep track of the reported bugs. The users and developers of software, report all the software related issues to bug tracking repository where the bug reports are further analyzed by one of the designated team members such as bug triager, senior developer or project manager of the bug repository. The validated bugs are then assigned by the bug triager to a suitable developer who make changes in source code to successfully resolve the bug. This successful resolving leads to a robust software product.

The following sections of this chapter focuses on the development of bug handling techniques for service sector softwares. The topic of bug fixing is broad to the point of multi-disciplinarity as it requires strong background knowledge of various disciplines such as software engineering, data mining, information retrieval, text analytics, mining software repositories, optimization techniques and statistical modeling. This chapter is designed in order to present a self-contained overview of research work in bug handling process and to allow readers to quickly gather the most recent and influential developments in bug resolving. The next part of this chapter explains the preliminaries related to software development and bug fixing process. In the later sections, bug handling process has been classified into three phases and various techniques used in each phase have been described. Finally, the chapter concludes by summarizing the most popular bug handling approaches and presenting some interesting future research directions related to optimization techniques in bug fixing.

BACKGROUND

This section provides the basic information related to software development and bug handling. First the different phases of software development life cycle are described followed by basic terminology, structure and life cycle of bug. Further different special categories of bug reports are presented. These special bugs are usually software context specific and may or may not appear in each software product.

18 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/bug-handling-in-service-sector-software/252117

Related Content

Data Accuracy Considerations with mHealth
Zaid Zekiria Sako, Vass Karpathiou, Sasan Adibiand Nilmini Wickramasinghe (2020). *Cognitive Analytics: Concepts, Methodologies, Tools, and Applications (pp. 1623-1638).*
www.irma-international.org/chapter/data-accuracy-considerations-with-mhealth/252102

Feature Selection Algorithms for Classification and Clustering
Arvind Kumar Tiwari (2020). *Cognitive Analytics: Concepts, Methodologies, Tools, and Applications (pp. 422-442).*
www.irma-international.org/chapter/feature-selection-algorithms-for-classification-and-clustering/252037

Data Mining Approach for Predicting the Likelihood of Infertility in Nigerian Women
Peter Adebayo Idowu, Jeremiah Ademola Balogunand Olumuyiwa Bamidele Alaba (2020). *Cognitive Analytics: Concepts, Methodologies, Tools, and Applications (pp. 1001-1027).*
www.irma-international.org/chapter/data-mining-approach-for-predicting-the-likelihood-of-infertility-in-nigerian-women/252067

Introduction of ACTA in Educational Settings
 (2025). *Qualitative Insights Through Applied Cognitive Task Analysis (pp. 67-90).*
www.irma-international.org/chapter/introduction-of-acta-in-educational-settings/370581

Quantitative Semantic Analysis and Comprehension by Cognitive Machine Learning
Yingxu Wang, Mehrdad Valipourand Omar A. Zatarain (2020). *Cognitive Analytics: Concepts, Methodologies, Tools, and Applications (pp. 673-688).*
www.irma-international.org/chapter/quantitative-semantic-analysis-and-comprehension-by-cognitive-machine-learning/252051