Journal of Database Management
Volume 33 • Issue 1

A Quantitative Function for Estimating the Comparative Values of Software Test Cases

Yao Shi, University of North Carolina Wilmington, USA*
Mark L. Gillenson, University of Memphis, USA
Xihui Zhang, University of North Alabama, USA

ABSTRACT

Software testing is becoming more critical to ensure that software functions properly. As the time, effort, and funds invested in software testing activities have been increased significantly, these resources still cannot meet the increasing demand of software testing. Managers must allocate testing resources to the test cases effectively in uncovering important defects. This study builds a value function that can quantify the relative value of a test case and thus play a significant role in prioritizing test cases, addressing the resource constraint issues in software testing and serving as a foundation of AI for software testing. The authors conducted a Monte Carlo simulation to exhibit application of the final value function.

KEYWORDS

Case Study, Resource Constraint, Simulation, Software Testing

INTRODUCTION

As software applications permeate everywhere in the world, people are becoming more sensitive to the validity and reliability of software applications (Juristo et al., 2006). Defects in software applications may result in tremendous financial loss and even innocent death (Felderer & Ramler, 2014). According to a report by Tricentis, a leading company providing software testing solutions, about 606 major software failures from 314 companies occurred throughout the world in 2017. These failures caused $1.7 trillion in financial losses, resulted in 268 years of cumulative unplanned downtime, and affected 3.6 billion people (Tricentis, n.d.). Billions of dollars are invested in software development every year around the world, and approximately 50 percent of the total elapsed time and more than 50 percent of the total costs are expended in testing the program or system being developed in a typical software development project (Boehm & Papaccio, 1988; Hailpern & Santhanam, 2002; Harrold, 2000; Myers et al., 2011). Despite the fact that software testing has been considered as an important phase in the software development life cycle to assure software quality, defects still cannot be entirely eradicated due to inadequate testing (Tricentis, n.d.; Whittaker, 2000).

Inadequate software testing usually results in defective applications and negative outcomes. Inspired by Whittaker's (2000) study, we identified five primary software testing practical issues

DOI: 10.4018/JDM.299559 *Corresponding Author

(untested code, untested combinations of input values, untested paths, untested operating environments, and defective testing procedures) that cause inadequate software testing. These five testing issues are rooted in resource constraints and technical constraints in software testing. On the one hand, a company might not have adequate resources such as budget, time, or personnel to run sufficient tests. On the other hand, a company might have enough resources but not have key technical support such as sophisticated algorithms, powerful testing tools, or expert testing engineers, resulting in defective testing procedures. Considering that most practical software testing issues result from resource constraints, we therefore focus on exploring the comparative value of software test cases in this study. The issue is that in a resource constrained testing environment, it is imperative for companies to choose the most valuable test cases to make their testing efforts as effective and efficient as possible. We believe that determining the most valuable test cases will help alleviate the resource constraints issues and thus dramatically improve the testing process.

Given that exhaustive software testing is impossible (Myers et al., 2011), maximizing the efficiency and effectiveness of software testing with limited resources becomes an important question in the software testing domain (Juristo et al., 2006). Although researchers have attempted to solve the problem of resource constraints in software testing from different perspectives in the last few decades (e.g., Biffl et al., 2006; Boehm, 2006; Felderer & Ramler, 2014; Wohlin & Aurum, 2006), the shortcomings of the prior studies indicate that the existing approaches cannot appropriately address the problem. This is because the software testing methods either lag behind software development methods or just take into account software engineering factors which cannot provide adequate guidance for improving software testing (Juristo et al., 2006; Talby et al., 2006).

Therefore, the research objective of this study is to explore a new mechanism involving the comparative value of test cases to increase the efficiency and effectiveness of software testing. This will be both a risk-based and cost-based approach to value estimation. Since test cases are at the core of software testing, all else being equal, choosing the highest value test cases can optimize software testing in a resource constrained environment (Biffl et al., 2006). To address the research objective, we intend to develop a function that assigns a relative value to a test case for the purpose of comparing it with the values of other test cases. This will result in the choice of the most effective test cases for a single application or the most effective set of test cases across a set of applications. We therefore initiate the research question surrounding the evaluation of software test cases: What is the relative value of a functional test case in software testing? It is important to point out that this is groundbreaking research. To our knowledge, no one has ever before attempted to assign comparative risk and cost-based numerical values to software test cases for the purpose of choosing the most effective and efficient test cases in a resource constrained testing environment. Furthermore, the quantitative function for assigning a comparative numerical value to a test case that we have developed was derived in a novel fashion from qualitative research.

RESEARCH FOUNDATION

Nature of Value in Software Test Cases

First, as a part of software testing, designing test cases is about creating inputs and predicting their associated outputs (Hass, 2014). Test cases create value in mitigating risks which can emerge as program defects. One can argue that some test cases are more valuable than others. For example, if a program is designed to be executed in different operating systems, the test cases for testing such compatibility are more valuable than those that work in a single operating system (Cohen et al., 2003).

Second, software testing, including the creation of test cases, is not free of charge. All companies attempt to decrease cost (e.g., software testing cost, cost of creating test cases) and increase benefit (e.g., application accuracy and reliability) when they develop a program. Therefore, the value of