Journal of Database Management
Volume 31 • Issue 2 • April-June 2020

Use of Qualitative Research to Generate a Function for Finding the Unit Cost of Software Test Cases

Mark L. Gillenson, University of Memphis, Memphis, USA
Thomas F. Stafford, Louisiana Tech University, Ruston, USA
Xihui "Paul" Zhang, University of North Alabama, Florence, USA
Yao Shi, University of Memphis, Memphis, USA

ABSTRACT

In this article, we demonstrate a novel use of case research to generate an empirical function through qualitative generalization. This innovative technique applies interpretive case analysis to the problem of defining and generalizing an empirical cost function for test cases through qualitative interaction with an industry cohort of subject matter experts involved in software testing at leading technology companies. While the technique is fully generalizable, this article demonstrates this technique with an example taken from the important field of software testing. The huge amount of software development conducted in today's world makes taking its cost into account imperative. While software testing is a critical aspect of the software development process, little attention has been paid to the cost of testing code, and specifically to the cost of test cases, in comparison to the cost of developing code. Our research fills the gap by providing a function for estimating the cost of test cases.

KEYWORDS

Case Research, Functional Test Case, Qualitative Generalization, Software Testing, Software Testing Costs

INTRODUCTION

This is a demonstration of case research used to generate an empirical function, and it is an unusual contribution to the literature in that respect. Mathematical expressions of functionality are usually developed statistically, supported with survey or simulation data (Akhavein et al., 1997; Kurfman et al., 2003). It is rare for a case study to be quantitative, yet mathematical functions can (conceivably) be derived in this way since deductions involving mathematical propositions are a subset of formal logic and nothing in formal logic is tied to a specific methodology (Lee, 1989). The evidence that is considered in case studies can be qualitative, quantitative, or both (Eisenhardt, 1989), but the primary purpose of a case study is to generate theory (Yin, 2013) and a theoretical representation of cause and effect (as is typically found in a statement of mathematical logic, such as a response function) is not out of the question in case research, just rare (Lee, 1989). It is, practically speaking, more likely that

DOI: 10.4018/JDM.2020040103

Copyright © 2020, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

response equations would lead to case studies (Hengl et al., 2007), since many statistical techniques benefit greatly from a qualitative interpolation process. In contrast to the conventional wisdom on case study uses and outcomes, however, we utilize an in-depth case study of industry experts to directly induce an empirical function through qualitative generalization (e.g., Yin, 2013).

Our methodology demonstrates the derivation of the variables and interrelated functionality of a candidate empirical expression. This methodology can be used in any situation in which there is a need to develop an empirical expression from case-based, qualitative research. While we will describe our innovative methodology in general terms, we believe that a detailed exposition of its actual use is essential to both understanding the process and to demonstrating its worth.

The actual example we have chosen is that of developing a function to estimate the cost of developing and executing a test case in the software testing process. The outcome of this process results in the description of the cost of generating and utilizing a functional test case for software testing purposes; this outcome of our case research is much in the way that Yin (2013) and Eisenhardt (1989) each suggests extracting theoretical meaning from case research. In that sense, the contribution of this study lies in generalizing from qualitative inquiry to a theoretical construction, as Yin and Eisenhardt would put it. For clarity, this will necessitate our providing some explanation of the elements of generating and executing software test cases, and of the software testing process, itself.

Case studies, concentrated as they are on a focused and bounded phenomenon embedded in a context, are particularly useful for identifying context-specific meaning (Eisenhardt, 1989; Miles et al., 2013). We consider that case study methods are typically intended to lead to theoretical developments as their own unique modality of generality (Yin, 2013), rather than the broad extension of generality to other contextual areas as is so typically the case with quantitative research. In that manner, we consider this in-depth investigation of software testing cost factors to be a revelatory approach (Sarker et al., 2012; Yin, 2013) meant for specific and detailed understanding of a unique context.

In addition, one of the most important potential theoretical contributions that could be made in the software development literature would be a functional model useful for determining the costs components of software testing, and this is an additional contribution of this study: the derivation of an empirical cost function as a generalization from an interpretive case study of software testing engineers brought together in a cohort as part of a certification program on software testing.

RESEARCH BACKGROUND

Importance of the Software Testing Problem

With the pervasiveness of software in most human activities, software defects may result in tremendous monetary and human loss. For example, in April 2015, a Bloomberg terminal in London crashed due to a technical issue in a third-party platform supplier. The shutdown forced the British Treasury to postpone 3 billion pounds (about USD $4.46 billion), in a short-term debt sale (Popper & Gough, 2015). Then in March 2018, a pedestrian was killed by an Uber's self-driving car in Tempe, Arizona. The self-driving software detected the pedestrian but incorrectly classified her as a "false positive" and decided the self-driving car did not need to stop (Lee, 2018). This fatal crash also resulted in Uber's self-driving program being stopped in North American where a large amount of money had been invested. There are numerous other cases resulting from software defects which ultimately could be attributed to insufficient software testing.

Software testing is a critical and an integral aspect of software development (Batra et al., 2016; Whitaker et al., 2012; Zhang et al., 2013), designed to find and correct the errors that arise in the software development process (Khan, 2010). However, the dilemma is that software cannot be tested exhaustively given testing resource constraints. Moreover, testing is increasingly costly, as a direct function of the increased complexity of software that is produced (Subramanian et al., 2017), yet there is little research that objectively demonstrates the way in which testing, costly as

20 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/article/use-of-qualitative-research-to-generate-a-function-for-finding-the-unit-cost-of-software-test-cases/249170

Related Content

Reuse a Repository of Conceptual Schemas in a Large Scale Project
Carlo Batini, Manuel F. Farasiand Riccardo Grosso (2006). *Advanced Topics in Database Research, Volume 5 (pp. 170-186).*
www.irma-international.org/chapter/reuse-repository-conceptual-schemas-large/4392

Simplifying the Formulation of a Wide Range of Object-Oriented Complex Queries
Reda Alhajj (2000). *Journal of Database Management (pp. 20-29).*
www.irma-international.org/article/simplifying-formulation-wide-range-object/3250

Data Mining: New Arsenal For Strategic Decision-making
Sufi M. Nazemand Bongsik Shin (1999). *Journal of Database Management (pp. 39-42).*
www.irma-international.org/article/data-mining-new-arsenal-strategic/51212

Possibility Theory in Protecting National Information Infrastructure
Richard Baskervilleand Victor Portougal (2005). *Advanced Topics in Database Research, Volume 4 (pp. 325-340).*
www.irma-international.org/chapter/possibility-theory-protecting-national-information/4381

Information Extraction from Free-Text Business Documents
Witold Abramowiczand Jakub Piskorski (2003). *Effective Databases for Text & Document Management (pp. 12-23).*
www.irma-international.org/chapter/information-extraction-free-text-business/9202