UNIVERSITI PUTRA MALAYSIA

ANALYSIS OF EXTRACTION COST OF QUALITY AND TESTING PHASE BY COMBINING SALLEH AND PRIMANDARIA'S MODEL

SHAIFUL FARITH BIN AHMAD

FSKTM 2018 58

ANALYSIS OF EXTRACTION COST OF QUALITY AND TESTING PHASE BY COMBINING SALLEH AND PRIMANDARIA'S MODEL

by

SHAIFUL FARITH BIN AHMAD

Thesis submitted to the School of Graduate Studies
Universiti Putra Malaysia
in fulfilment of the requirements for the
Master of Software Engineering

July 2018

ABSTRACT

Abstract of thesis presented to Universiti Putra Malaysia in fulfilment of the requirement for the Master of Software Engineering

by **Shaiful Farith Bin Ahmad**

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Software Testing activities are important to find defects, gain confidence about the level of quality, provide information for decision-making and prevent defects. Nowadays, testing from independent organisations is the best practice to measure the confidence level of stakeholders before deploying and using a system, especially high impact ones. However, currently, the government agencies of Malaysia do not have any cost estimation models to implement quality and testing phase by independent organisations.

Therefore, the purpose of this study is to analyse and extract the cost of quality and testing phase from the total cost of a software project. The objectives of this study are to design an extraction model to extract the cost of quality and testing phase from the total cost of a software project, to build a prototype for the model and to evaluate it.

This study provides options to extract the cost of quality and testing phase from the total cost of a software project. The options will be based on estimation and/or prediction. Constructive Cost Model (COCOMO) and estimation model by Saleh 2011 were chosen as estimation techniques, while linear regression model was chosen to predict the cost of quality and testing phase. This study used Function Point Analysis (FPA) to measure the size of system.

The best-fitting line that was developed based on four existing projects that implemented the outsourced test team was $\hat{y} = 92{,}774.32 + 216.04\ \hat{x}$, where the slope of the line (β) was 216.04, and the intercept (α) was 92,774.32. This study had validated the predicted linear regression by Mean Magnitude of Relative Error (MMRE) and PRED (0.25). The result for MMRE was 0.15 and PRED (0.25) was 1. A small value of MMRE means the estimation is acceptable and PRED (0.25) of 1 means the Prediction Quality is acceptable.

Keywords: software testing; software cost estimation, function point; COCOMO; linear regression.

ABSTRAK

Abstrak tesis yang dikemukakan kepada Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Sarjana Kejuruteraan Perisian

Oleh **Shaiful Farith Bin Ahmad**

FAKULTI SAINS KOMPUTER DAN TEKNOLOGI MAKLUMAT

Aktiviti pengujian perisian adalah penting dalam menjejaki ralat sistem, meningkatkan tahap keyakinan terhadap kualiti sistem, membekalkan maklumat bagi memudahkan sesuatu keputusan dibuat, dan mengelakkan sistem berlakunya ralat. Fasa pengujian dan kualiti daripada pihak ketiga yang berkecuali pada masa kini adalah amalan terbaik untuk meningkatkan tahap keyakinan pemegang kepentingan sebelum sistem dipasang dan digunakan khususnya bagi sistem yang berimpak tinggi. Walau bagaimanapun, pada masa kini, agensi kerajaan Malaysia tidak mempunyai model bagi menganggarkan kos fasa pengujian dan kualiti oleh pihak ketiga.

Oleh itu, tujuan projek ini adalah untuk menganalisa dan ekstrak kos fasa pengujian dan kualiti daripada kos keseluruhan projek perisian. Objektif projek adalah untuk merekabentuk model bagi mengekstrak kos fasa pengujian dan kualiti daripada kos keseluruhan projek perisian, membangunkan prototaip untuk model pengekstrakan kos fasa pengujian dan kualiti, dan membuat penilaian terhadap prototaip tersebut.

Projek ini menyediakan pilihan untuk ekstrak kos fasa pengujian dan kualiti daripada kos keseluruhan projek perisian. Pilihan tersebut berdasarkan anggaran dan/atau ramalan. Constructive Cost Model (COCOMO) dan anggaran model oleh Saleh 2011 telah dipilih untuk pilihan teknik anggaran. Manakala model regresi linear telah dipilih untuk meramal kos fasa pengujian dan kualiti. Projek ini telah menggunakan Function Point Analysis (FPA) sebagai asas untuk mengukur saiz sistem.

Regresi linear terbaik telah dihasilkan berdasarkan kepada 4 projek yang telah melaksanakan pasukan pengujian luar iaitu "$\hat{y} = 92{,}774.32 + 216.04\ \hat{x}$", di mana kecerunan garisan (β) adalah 216.04, dan pintasan (α) adalah 92,774.32. Projek ini telah validasi regrasi linear dengan Magnitude of Relative Error (MMRE) dan PRED (0.25). Keputusan MMRE adalah 0.15 dan PRED (0.25) adalah 1. Nilai kecil bagi MMRE menunjukkan hasil anggaran tersebut boleh diterima dan PRED (0.25) adalah 1, bermaksud Kualiti Ramalan juga boleh diterima.

Kata kunci: pengujian perisian, anggaran kos perisian, nilai saiz fungsi sistem, model anggaran kos perisian, regresi linear

ACKNOWLEDGEMENT

In the name of Allah, The Most High, The Most Gracious, The Most Merciful. All praise and thanks are due to Allah: for His help and instilling in me the courage and strength to complete of my dissertation. Nothing is possible without His help.

First of all, I want to express my gratitude and thanks to my supervisor, Dr. Pathiah Abd Samat for his guidance and supervision. I am also expressing my gratitude to all lecturers from Faculty of Computer Science and Information Technology (CSIT) and course mate Master of Software Engineering batch of 2016/2017.

I am very fortunate and grateful for the support of the Malaysian Testing Center of Excellence (MyTCoE) team in making this thesis successful. Not forgetting also to fellow friends who have been so supportive along the way of doing my thesis.

Not least of all, I also would like to express my wholehearted thanks to my family for their undying support, their unwavering belief that I can achieve so much. Because of their unconditional love and prayers, I have the chance to complete this thesis

Last but not least, deepest thanks go to all people who took part in making this thesis real.

DECLARATION

I hereby confirm that:

- this thesis is my original work;
- quotations, illustrations and citations have been duly referenced;
- this project report has not been submitted previously or concurrently for any other degree at any other institutions;
- intellectual property from the project report and copyright of project report are fully-owned by Universiti Putra Malaysia, as according to the Universiti Putra Malaysia (Research) Rules 2012;
- written permission must be obtained from supervisor before project report is published (in the form of written, printed or in electronic form);

Signature: ______________________ Date: ________________

Name and Matric No.: **SHAIFUL FARITH BIN AHMAD**

GS46574

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figures		Page

Figures		Page

LIST OF EQUATION

CHAPTER 1

INTRODUCTION

This chapter will explain the importance of software testing, benefits of early testing and level of independent testing.

1.1. Importance of Software Testing

The World Quality Report 2014-2015 stated that the proportion of the IT budget allocated to Quality Assurance (QA) and testing has grown from 18% in 2012 and continues to increase, reaching 29% by 2017. The rise in the proportion of corporate IT spend on QA and testing is linked to the growing importance of application quality, driven by digital transformation initiatives (Capgemini, HP, and Sogeti 2014).

Figure 1.1 Importance of software testing

The increase in spending shows that the testing activities are getting more important and are given more attention as illustrated in Figure 1.1. Generally, the testing activities are crucial to find defects early, gain confidence of stakeholders about the level of quality before deploying and using the system, provide information for decision-making; either the system is ready to be used or need to be tested again, and prevent defects during maintenance phase (Graham et al. 2006).

1.2. Early Testing in Software Development Life Cycle

Testing activities are recommended to start as early as possible during the Software Development Life Cycle (SDLC) stage as one of the General Testing Principles is early testing. Samaroo, Thompson, and Hambling (2015) developed a comparative cost to correct errors in the SDLC stage in Table 1.1. The table shows that it is important to implement testing activities as early as possible as it will reduce the cost to correct the errors.

Table 1.1 Comparative cost to correct errors

Stage error is found	Comparative cost
Requirements	$1
Coding	$10
Program testing	$100
System testing	$1,000
User acceptance testing	$10,000
Live running	$100,000

The Cost Escalation Model can be generated from the comparative cost table to correct errors. Figure 1.2 shows the effect of identification time on the cost of errors. It shows that the earlier an error is found, the less it costs to correct it.

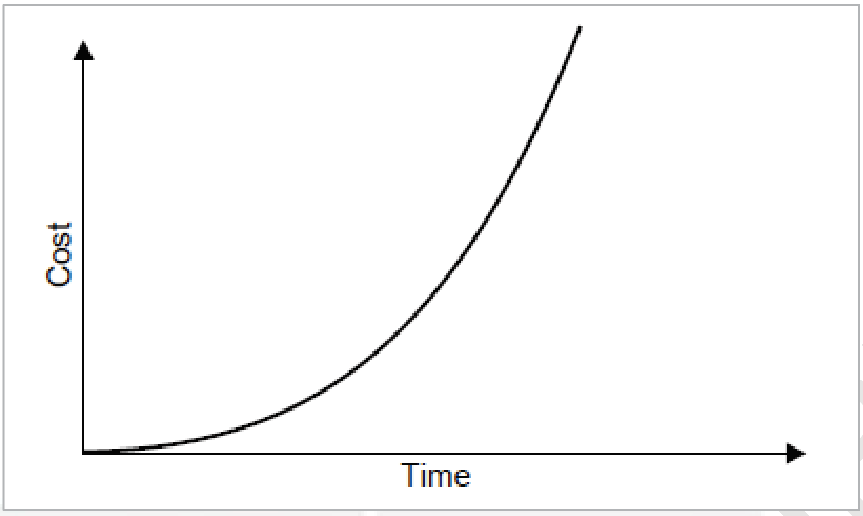

Figure 1.2 Effect of identification time on the cost of errors

1.3. Level of Independent Testing

Implementation and execution of testing can be done by using the six levels of independent testing as shown in Figure 1.3. The lowest level in the independent testing is the developer, which is the testing executed by the developer itself. The highest level in the independent testing is the outsourced test team or tester, e.g., contractor or other organisations (Samaroo, Thompson, and Hambling 2015).

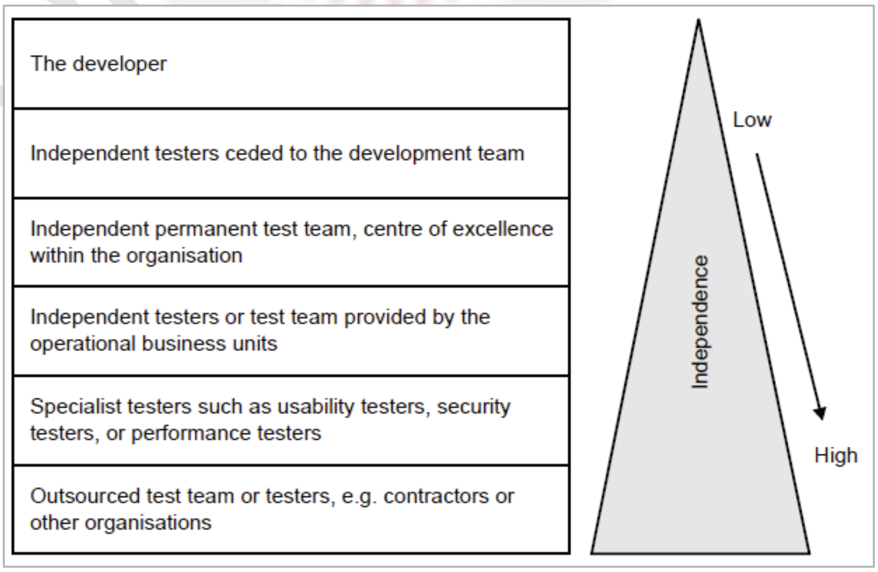

Figure 1.3 Levels of independent testing

Nowadays, implementing quality and testing phase from an outsourced test team is the best practice to measure the confidence level of stakeholders before the system is deployed and used, especially for high impact system. An outsource test team needs to be free from the management, finance and development teams. Practically, the types of testing in the quality and testing phase are not limited to Functional Testing, Performance Testing, Stress Test, Load Test, and Security.

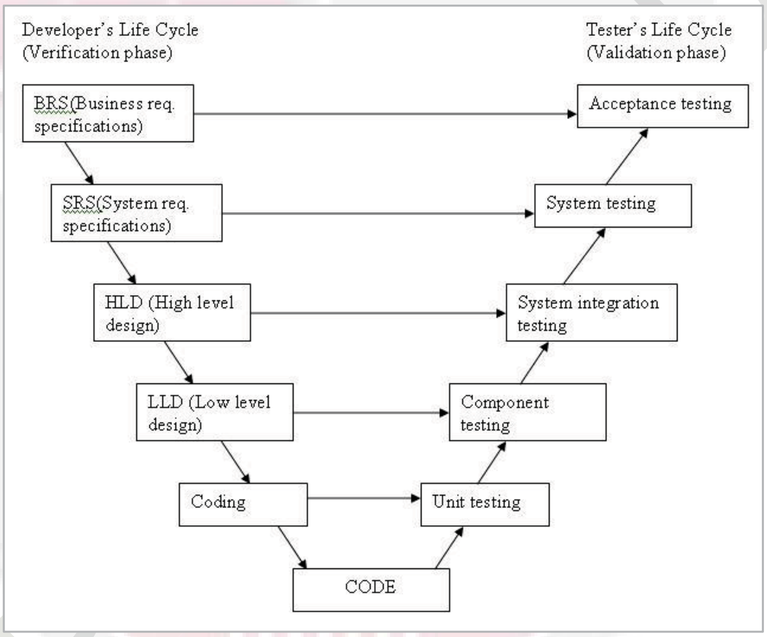

Figure 1.4 Diagram of V-model

The implementation of quality and testing phase by an outsource test team suits the V-Model, as it ensures that all levels are tested. In software development, the V-model, from the International Software Testing Qualifications Board (ISTQB) shown in Figure 1.4, represents a development process that may be considered as an extension of the waterfall model and a more general V-model. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape.

REFERENCES

Agrawal, Priya, and Shraddha Kumar. 2016. “Early Phase Software Effort Estimation Model.”

Ahn, Yunsik, Jungseok Suh, Seungryeol Kim, and Hyunsoo Kim. 2003. “The Software Maintenance Project Effort Estimation Model Based on Function Points.” *Journal of Software Maintenance and Evolution* 15(2): 71–85.

Algabri, Mohammed, Fahman Saeed, Hassan Mathkour, and Nejmeddine Tagoug. 2015. “Optimization of Soft Cost Estimation Using Genetic Algorithm for NASA Software Projects.” : 3–6.

Banker, Rajiv D, and Sandra A Slaughter. 1999. 15213 Complexity *The Economics of Software Quality Practices*.

Capgemini, HP, and Sogeti. 2014. *World Quality Report 2014-15: Sixth Edition*. https://www.capgemini.com/resources/world-quality-report-2014-15/.

Dehghan, Ali et al. 2017. “Predicting Likelihood of Requirement Implementation within the Planned Iteration : An Empirical Study at IBM.” : 384–94.

Ebert, C.a, and H.b Soubra. 2014. “Functional Size Estimation Technologies for Software Maintenance.” *IEEE Software* 31(6): 24–29. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910680475&partnerID=40&md5=9d655a48da65e262428e3f9aee211b2c.

Federal Student Aid, U.S. Department of Education. 2013. “Independent Verification & Validation (IV&V) Handbook.” http://federalstudentaid.ed.gov/static/gw/docs/ciolibrary/FSA-IVV-Handbook.pdf.

Fessi, Boutheina A, Yosra Miaoui, and Noureddine Boudriga. 2013. “A Managerial Issues-Aware Cost Estimation of Enterprise Security Projects.” : 1–6.

Ghosh, Sanjay, Srini Ramaswamy, and Raoul Praful Jetley. 2013. “Towards Requirements Change Decision Support.”

Graham, Dorothy, Erik van Veenendaal, Isabel Evans, and Rex Black. 2006. “FOUNDATIONS OF SOFTWARE TESTING: ISTQB CERTIFICATION.”

Hailpern, B., and P Santhanam. 2002. “Software Debugging, Testing, and Verification.” 41(1).

Hunt, Bob, Tony Abolfotouh, and John Carpenter. 2014. "Software Test Costs and Return on Investment (ROI) Issues." (March): 1–25.

Leblanc, Richard, Adair Dingle, Jon Duncan Hagar, and John Knight. 2015. *Software Metrics Fundamentals of Dependable Computing for Software Engineers*.

Longstreet, D. 2005. 2 Longstreet Consulting Inc. Accessed *Function Points Analysis Training Course*. http://www.poli.usp.br/d/pmr2490/fp_training.pdf.

Ministry of Finance, Malaysia. 2013. "Kaedah Perolehan Kerajaan PK 2.1." http://1pp.treasury.gov.my/topik/fail/5/muat-turun.

Mishra, Siba. 2014. "Estimating Development Size and Effort of Business Process Service-Oriented Architecture Applications." (Icsai): 1006–11.

Norman E.Fenton, Shari Lawrence Pfleeger. 1998. "Software Metrics_ A Rigorous and Practical Approach, Revised."

Patil, Lalit V., Nitin M Shivale, S. D. Joshi, and V. Khanna. 2014. "IMPROVING THE ACCURACY OF CBSD E FFORT.": 1385–91.

Patil, Lalit V, and Rina M Waghmode. 2014. "GENERIC MODEL OF SOFTWARE COST ESTIMATION : A HYBRID APPROACH." : 1379–84.

Pfahl, Dietmar, and Marlon Dumas. 2014. "Software Economics – Fall 2014 Workshop 2-3 : Function Point Analysis." https://courses.cs.ut.ee/MTAT.03.244/2014_fall/uploads/Main/workshop2.pdf.

Pow-Sang, José Antonio, and Ricardo Imbert. 2012. "Effort Estimation in Incremental Software Development Projects Using Function Points." *Communications in Computer and Information Science* 340 CCIS: 458–65.

Primandaria, Putu Linda, and Sholiq. 2015. "Effort Distribution to Estimate Cost in Small to Medium Software Development Project with Use Case Points." *Procedia Computer Science* 72: 78–85. http://dx.doi.org/10.1016/j.procs.2015.12.107.

Qi, Fumin et al. 2017. "Software Effort Estimation Based on Open Source Projects: Case Study of Github." *Information and Software Technology* 92: 145–57. http://linkinghub.elsevier.com/retrieve/pii/S0950584917302239.

Rajewski, Bob. 2017. "The Importance of Independent Verification and Validation of Software in The Medical Device Industry." http://www.critech.com/files/122876213.pdf.

Saleh, Kassem. 2011. “Effort and Cost Allocation in Medium to Large Software Development Projects.” *International Journal of Computers (1)* 5(1): 74–79.

Samaroo, Angelina, Geoff Thompson, and Brian Hambling. 2015. *Software Testing: An ISTQB-BCS Certified Tester Foundation Guide*. Third. ed. Brian Hambling.

Schett, Nancy Merlo. 2003. “Constructive Cost Model (COCOMO).”

Sequeira, Sérgio, and Eurico Lopes. 2015. “Simple Method Proposal for Cost Estimation from Work Breakdown Structure.” *Procedia Computer Science* 64: 537–44. http://dx.doi.org/10.1016/j.procs.2015.08.559.

Shahwaiz, Syed Ahmad, Ali Afzal Malik, and Nosheen Sabahat. 2016. “A Parametric Effort Estimation Model for Mobile Apps.”

Shani, Guy, and Asela Gunawardana. 2011. “Evaluating Recommendation Systems.” *Recommender Systems Handbook*: 257–97. http://link.springer.com/10.1007/978-0-387-85820-3_8.

Shields, Anne. 2014. Market Realist *Enterprise Software*. http://marketrealist.com/analysis/stock-analysis/technology/enterprise-software/charts/?featured_post=72797&featured_chart=72031.

Tintarev, Nava, and Judith Masthoff. 2012. “Evaluating the Effectiveness of Explanations for Recommender Systems: Methodological Issues and Empirical Studies on the Impact of Personalization.” *User Modeling and User-Adapted Interaction* 22(4–5): 399–439.

Toka, Derya. 2013. “Accuracy of Contemporary Parametric Software Estimation Models : A Comparative Analysis.” : 313–16.

Wohlin, Claes et al. 2000. *Experimentation in Software Engineering: An Introduction*. ed. Victor R. Basili. SPRINGER SCIENCE+BUSINESS MEDIA, LLC.