



**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**

All material contained within the thesis, including without limitation text, logos, icons, photographs and all other artwork, is copyright material of Universiti Putra Malaysia unless otherwise stated. Use may be made of any material contained within the thesis for non-commercial purposes from the copyright holder. Commercial use of material may only be made with the express, prior, written permission of Universiti Putra Malaysia.

Copyright © Universiti Putra Malaysia



## 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 ( $\beta$ ) was 216.04, and the intercept ( $\alpha$ ) 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 ( $\beta$ ) adalah 216.04, dan pintasan ( $\alpha$ ) 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

	<b>Page</b>
<b>ABSTRACT</b>	iii
<b>ABSTRAK</b>	iv
<b>ACKNOWLEDGEMENT</b>	v
<b>DECLARATION</b>	vi
<b>LIST OF TABLES</b>	ix
<b>LIST OF FIGURES</b>	xii
<b>LIST OF EQUATION</b>	xv
<b>CHAPTER</b>	
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Importance of Software Testing	1
1.2 Early Testing in Software Development Life Cycle	2
1.3 Level of Independent Testing	3
<b>2 PROBLEM DEFINITION, OBJECTIVE, AND SCOPE</b>	<b>5</b>
2.1 Definition of The Problem	5
2.1.1 The Topic	5
2.1.2 The Research Problem	6
2.1.3 Background and Justification	6
2.1.4 Deficiencies in the Evidence	8
2.1.5 Audience	8
2.2 The Objective of The Study	9
2.3 The Scope of The Study	9
<b>3 LITERATURE REVIEW</b>	<b>11</b>
3.1 Importance of Independent Testing	11
3.2 Function Points Analysis (FPA)	13
3.3 Constructive Cost Model (COCOMO)	14
3.4 Extract Cost of Quality and Testing	16
3.5 Measures of Dependency	20
<b>4 METHODOLOGY</b>	<b>22</b>
4.1 Planning Phase	22
4.2 Analysis Phase	23
4.3 Design Phase	23
4.4 Implementation Phase	24
4.5 Evaluation Phase	24
4.6 Conclusion	24



<b>5</b>	<b>DEVELOPMENT</b>	<b>25</b>
5.1	Analysis Phase	25
5.1.1	Analysis on Calculate Function Point	26
5.1.2	Analysis of Estimate Effort and Cost	32
5.1.3	Analysis for Predict Cost of Quality and Testing	39
5.2	Design Phase	40
5.2.1	Conceptual Design	40
5.2.2	Physical Design	45
5.3	Implementation Phase	50
5.3.1	Implement 'Calculate Function Point'	50
5.3.2	Implement 'Estimate Effort and Cost'	52
5.3.3	Implement 'Predict Cost of Quality and Testing'	55
<b>6</b>	<b>EVALUATION</b>	<b>59</b>
6.1	Mean Magnitude of Relative Error (MMRE)	59
6.2	Prediction Quality, PRED (0.25)	60
6.3	Evaluate by Survey	60
<b>7</b>	<b>RESULTS AND DISCUSSION</b>	<b>64</b>
7.1	Result of Mean Magnitude of Relative Error (MMRE)	64
7.2	Result of PRED (0.25)	66
7.3	Result of Evaluation by Survey	67
7.3.1	Graphical Presentation Result	67
7.3.2	Summary of Data Results	74
7.4	Discussion	79
7.4.1	Relationship between FPC and Cost of Quality and Testing	79
7.4.2	Result between Prediction Model with Estimation Model	80
7.4.3	Contribution of Study	80
7.4.4	Limitation of Study	81
<b>8</b>	<b>CONCLUSION</b>	<b>83</b>
8.1	Conclusion	83
8.2	Future Work	84
	<b>REFERENCES</b>	<b>87</b>
	<b>APPENDIX</b>	<b>90</b>

## LIST OF TABLES

<b>Table</b>		<b>Page</b>
Table 1.1	Comparative cost to correct errors	2
Table 3.1	Different level of independent testing	11
Table 3.2	Advantages of using Function Points Analysis	13
Table 3.3	COCOMO Model Coefficients by Algabri et al. 2015	14
Table 3.4	Advantages of COCOMO model	15
Table 3.5	Disadvantages of COCOMO model	16
Table 3.6	Percentage effort on software activities by Saleh 2011	19
Table 3.7	Percentage effort on software activities by Primandaria & Sholih 2015	20
Table 5.1	Description of FP components	27
Table 5.2	DET, RET, and FTR relationship with components of FP	28
Table 5.3	DET's rate and score for External Inputs	28
Table 5.4	DET's rate and score for External Outputs and External Inquiries	28
Table 5.5	DET's rate and score for Internal Logical Files	29
Table 5.6	DET's rate and score for External Interface Files	29
Table 5.7	Total Number of Unadjusted Function Points (UFPs)	30
Table 5.8	Degrees of influence scale	30
Table 5.9	General system characteristics	31
Table 5.10	Function Point Languages	33
Table 5.11	Levels of COCOMO	34
Table 5.12	Development modes of COCOMO	35
Table 5.13	Basic COCOMO Model Coefficients	35

<b>Table</b>		<b>Page</b>
Table 5.14	Estimation for effort and/or cost of quality and testing	36
Table 5.15	Percentages of effort and pay rate for software activities	37
Table 5.16	Median salary for 14 positions in software development	38
Table 5.17	Design for interaction style	41
Table 5.18	Sub-activities for main activities	43
Table 5.19	Object-action analysis	44
Table 5.20	Functions and relationship between functions	45
Table 5.21	Information requirements for task	45
Table 6.1	Likert scale for usability metric	61
Table 6.2	Survey questions related to ease of use of 'Calculate Function Point' feature	62
Table 6.3	Survey questions related to ease of use of 'Estimate Effort and Cost' feature	62
Table 6.4	Survey questions related to ease of use of 'Predict Cost of Quality and Testing' feature	62
Table 6.5	Survey questions related to ease of use of 'Predict FPC' feature	63
Table 7.1	Result of MRE by using Prediction Model	64
Table 7.2	Result of MRE by combining Salleh and Primandaria's Models	65
Table 7.3	Result of survey related to ease of use of 'Calculate Function Point' feature	74
Table 7.4	Result of survey related to ease of use of 'Estimate Effort and Cost' feature	75
Table 7.5	Result of survey related to ease of use of 'Predict Cost of Quality and Testing' feature	76

<b>Table</b>		<b>Page</b>
Table 7.6	Result of survey related to ease of use of 'Predict FPC' feature	78
Table 7.7	Result of MMRE and PRED (0.25)	80
Table 7.9	Scope of work for historical data project	82
Table 8.1	Examples of effort and cost for each software activity by Saleh (2011)	85
Table 8.2	Examples of schedule and effort distributions for each software activity by Saleh (2011)	85



## LIST OF FIGURES

<b>Figures</b>		<b>Page</b>
Figure 1.1	Importance of software testing	1
Figure 1.2	Effect of identification time on the cost of errors	3
Figure 1.3	Levels of independent testing	3
Figure 1.4	Diagram of V-model	4
Figure 2.1	Perolehan Kerajaan PK 2.1	7
Figure 2.2	Software testing in 11th Malaysia Plan	8
Figure 3.1	Relation between project size and project effort	21
Figure 4.1	Research Methodology	22
Figure 5.1	Step in the analysis phase	25
Figure 5.2	Step to calculate Function Points	26
Figure 5.3	Step in estimate effort and cost	32
Figure 5.4	Median salary for software developer in Malaysia	38
Figure 5.5	Step in predicting cost and number of function point	39
Figure 5.6	Low-fidelity prototype	42
Figure 5.7	High-fidelity prototype	42
Figure 5.8	Operational design for Calculate Function Point	47
Figure 5.9	Operational design for Estimate Effort and Cost	47
Figure 5.10	Operational design for Predict Cost of Quality and Testing	48
Figure 5.11	Use case diagram of prototype	49
Figure 5.12	Sequence diagram of prototype	49

<b>Figures</b>	<b>Page</b>	
Figure 5.13	UI for calculating Function Point Count	51
Figure 5.14	Script to calculate the total number of UFP	51
Figure 5.15	Script to calculate FPC	52
Figure 5.16	UI for estimate effort and cost	53
Figure 5.17	Script to estimate project effort, duration, and team members	53
Figure 5.18	Script to calculate effort for each software activity	54
Figure 5.19	Script to calculate the cost of each software activity	54
Figure 5.20	UI for predict cost of quality and testing	55
Figure 5.21	Definition linear regression for quality and testing	56
Figure 5.22	History data FPC and cost of quality and testing	56
Figure 5.23	Script to predict the cost of software activities	57
Figure 5.24	Script to predict of number of FP, project effort, duration, and team members	57
Figure 7.1	Calculation for MMRE by using Prediction Model	65
Figure 7.2	Calculation for MMRE by Combining Salleh and Primandaria's Models	65
Figure 7.3	Result of PRED (0.25) by using Prediction Model	66
Figure 7.4	Result of PRED (0.25) by Combining Salleh and Primandaria's Models	66
Figure 7.5	Respondent's background in term gender, age, certification in software testing	67

<b>Figures</b>		<b>Page</b>
Figure 7.6	Current positions of respondents	68
Figure 7.7	Years of working experience of respondents	69
Figure 7.8	Feedback from respondents to 'Calculate Function Point' feature	70
Figure 7.9	Feedback from respondents to 'Estimate Effort and Cost' feature	71
Figure 7.10	Feedback from respondents to 'Predict Cost of Quality and Testing' feature	72
Figure 7.11	Feedback from respondents to 'Predict FPC' feature	73
Figure 7.12	Graph of Relationship between FPC and Cost of Quality and Testing	79

## LIST OF EQUATION

Equation		Page
Equation 3.1	Equation of overall cost of the project	15
Equation 3.2	Equation of Test cost by Hunt, Abolfotouh, and Carpenter 2014	17
Equation 3.3	Equation of IV&V Cost by Hunt, Abolfotouh, and Carpenter 2014	17
Equation 3.4	Equation of linear regression	21
Equation 5.1	Equation of Value Adjustment Factor (VAF)	31
Equation 5.2	Equation of Function Point Count (FPC)	31
Equation 5.3	Equation of COCOMO Model	36
Equation 5.4	Equation of estimate number of team member and cost	36
Equation 6.1	Equation for Relative Error	59
Equation 6.2	Equation for Magnitude Relative Error	59
Equation 6.3	Equation of Prediction Quality	60



# 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*

<b>Stage error is found</b>	<b>Comparative cost</b>
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.

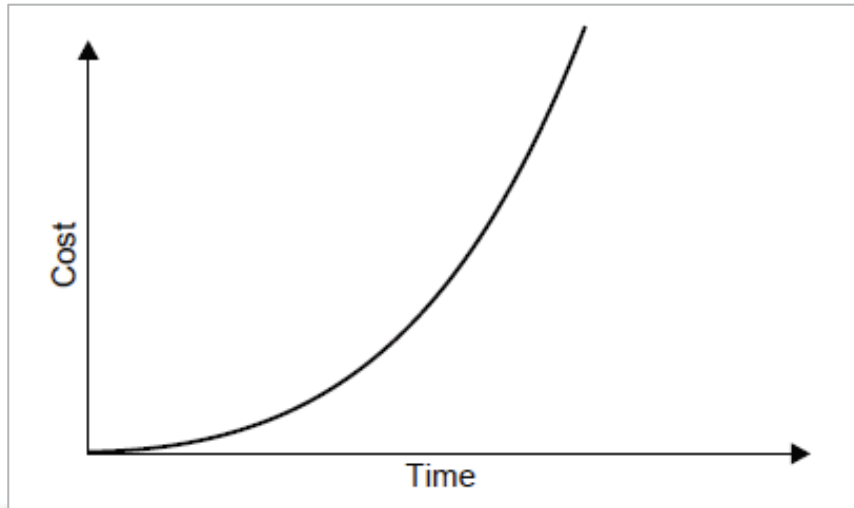


Figure 1.2 Effect of identification time on the cost of errors

### 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).

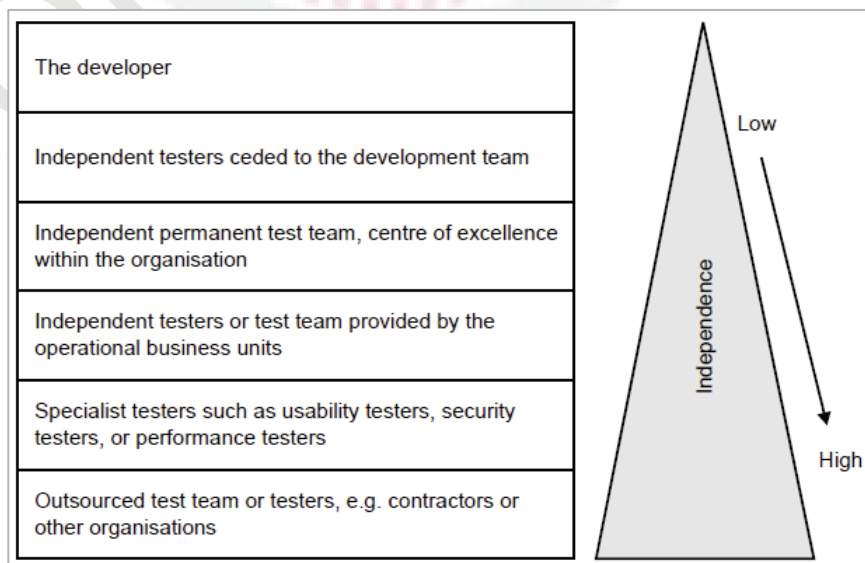


Figure 1.3 Levels of independent testing

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.

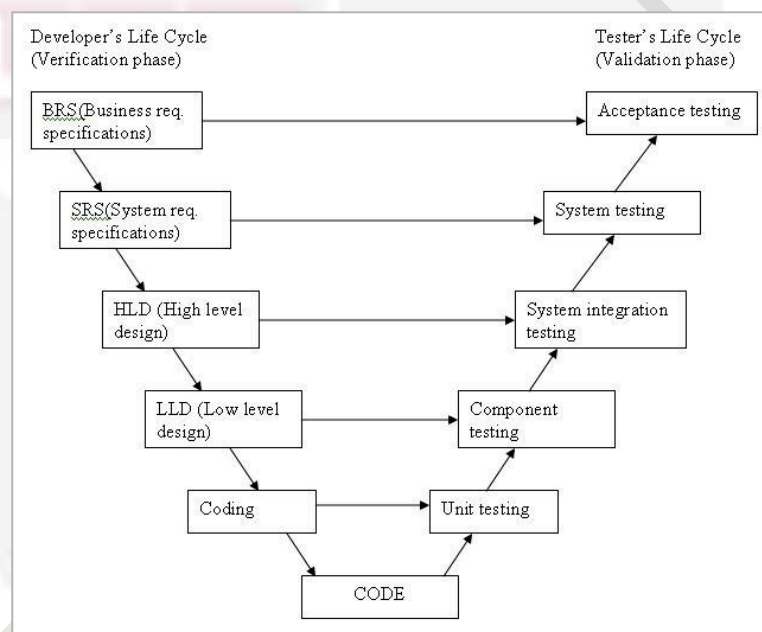


Figure 1.4 Diagram of V-model

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 Nouredine Boudriga. 2013. "A Managerial Issues-Aware Cost Estimation of Enterprise Security Projects." : 1–6.
- Ghosh, Sanjay, Sridhar 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](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](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 Sholih. 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](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](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.