

Ain Shams University
Faculty of Computer and Information Sciences
Computer Science Department



Faculty of Computer and Information Sciences

Cost, Time and Quality Trade-off in Software Engineering

A Dissertation Submitted to the Department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University; in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Science

By

Abdelelah Ghaleb Farhan Saif

Master of Computer Science, Faculty of Information Technology and Computer Science,
Yarmouk University, Jordan 2010

Supervised By

Prof. Dr. Zaki Taha Ahmed Fayed

Professor of Computer Science

Faculty of Computer and Information Sciences

Ain Shams University

Dr. Safia Abbas Mahmoud Abbas

Lecturer of Computer Science

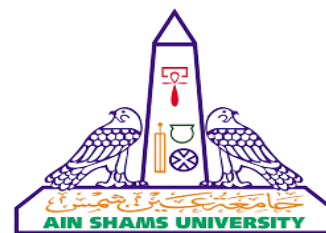
Faculty of Computer and Information Sciences

Ain Shams University

Cairo- 2017

Ain Shams University

**Faculty of Computer and Information Sciences
Computer Science Department**



Faculty of Computer and Information Sciences

Cost, Time and Quality Trade-off in Software Engineering

A Dissertation Submitted to the Department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University; in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Science

By

Abdelelah Ghaleb Farhan Saif

Signature of Author:

Committee

Signature & Date

Prof. Dr. Zaki Taha Ahmed Fayed (Supervisor)

Dr. Safia Abbas Mahmoud Abbas (Supervisor)

Prof. Dr. Said Ghoniemy (Examiner)

Prof .Dr. Atef Zaki Ghalwash (Examiner)

Cairo- 2017

DECLARATION

This part declares that the work reported in this dissertation was done by author to certify that:

1. The dissertation comprises only my original work toward the PhD,
2. Due acknowledgement has been made in the text to all other materials used,
3. The dissertation consists of 130 pages, exclusive of references and appendices.

Abdulelah Al-Mikhlafi

*2017
Cairo- Egypt*

CERTIFICATE

This is certified that the dissertation, entitled "**Cost, Time and Quality Trade-off in Software Engineering**" submitted by Mr. **Abdelelah Ghaleb Farhan Saif**; in partial fulfillment of the requirements for the degree of Doctor of Philosophy in computer science, faculty of computer and information sciences at Ain Shams University is a bonafide record of research work carried out by the candidate under our supervision and guidance. The work involved in this dissertation has not been previously submitted to this or other university for PhD. or any other degree. We further declare that the materials obtained from other sources have been duly acknowledged in the dissertation.

Prof. Dr. Zaki Taha Ahmed Fayed

Professor of Computer Science

Faculty of Computer and Information Sciences

Ain Shams University

Date: 17/ 01/2017

Signature:

Dr. Safia Abbas

Lecturer of Computer Science

Faculty of Computer and Information Sciences

Ain Shams University

Date: 17/ 01/2017

Signature:

ACKNOWLEDGMENT

First of all, I must glorify **Allah** the Almighty who gave me courage and patience to carry out this work successfully.

I would like to express my sincere gratitude to my supervisors, **Prof. Dr. Zaki Taha Ahmed Fayed** and **Dr. Safia Abbas Mahmoud Abbas**. They were always receptive to the ideas that I came up with, and provided me great knowledge and continuous encouragements and they always taught me how to explain the idea and how to move toward the goal.

I would like to dedicate this work to my family in Yemen who courage me to continue in education. Many thanks to **my parents, my wife, my brothers and my sisters**.

Big thank to **Yemeni government** which provides me the funding support to perform this work.

Finally, I want to thank **my friends** in Ain Shams University for their help.

Abdulelah Al-Mikhlafi

Cairo- 2017

Abstract

Cost, schedule and quality are highly related factors (objectives) in software development. They constitute the three sides of the triangle. It is hard to raise the quality without raising either the cost or schedule or both for the software under development. Similarly, development schedule cannot be decreased without sacrificing software product quality and/or raising development cost.

The Discrete Time, Cost and Quality Trade-off Problem (DTCQTP) is special case of this problem in which there are number of execution modes for each activity (modes can be either bids offered by subcontractors to develop an activity where each bid has duration, cost and quality), and the best execution mode of every activity should be determined in order to optimize the total cost, time and quality of the software project.

This work includes:

1. Introducing an optimized cost-quality model based on dataset in Constructive Cost Model (COCOMO) format by adapting Intelligent Water Drops (IWD) algorithm for optimizing COCOMO II Post Architecture (PA) and Constructive Quality Model (COQUAMO/COQUALMO) models to achieve more accurate estimation/prediction of software development effort and hence cost, time and quality on the project level and to allow making trade-off analysis. For this model, the prediction accuracy of IWD is compared with the original models, genetic algorithm (GA) and Problem Data Based Optimization for Continuous Optimization (PDBO-CO) which is developed for this purpose to enable project manager selects the suitable model. The experiment has been conducted on NASA 93 dataset.
2. Introducing a cost-quality model that optimizes effort and hence cost, time and quality (making trade-off analysis) by finding the best project options on the project level. For this model, a comparison is done among IWD, GA and Problem Data Based Optimization (PDBO) in terms of the stability of selecting the best project options, quality of solution and Processing Time (PT) to enable project manager selects the best one. The experiments are conducted on an imaginary software project with size of 25 Thousand Source Lines of Code (25KSLOC) and on Jet Propulsion Laboratory (JPL) flight software project.
3. Developing an application (solving DTCQTP). The algorithms that are used to solve such problem (DTCQTP) are IWD, GA, Ant Colony Optimization (ACO), Egyptian Vulture Optimization (EVO) and PDBO which is developed for this purpose. By using some

examples from the literature, the results of these algorithms are compared in terms of quality of solution and efficiency to enable project manager selects the best one.

4. Finally, comparing the proposed PDBO-CO with Intelligent Water Drops for Continuous Optimization (IWD-CO) for function optimization.

The results are as follow:

For the **optimized cost-quality based on NASA 93 dataset**, we conclude: Effort and time obtained by PDBO-CO, IWD, and GA are closer to the actual ones than that obtained by the original COCOMO II PA model. By doing a comparison among PDBO-CO, IWD, GA and the original COCOMO II PA model regarding prediction accuracy, the Mean Magnitude of Relative Error (*MMRE*) of PDBO-CO, and IWD for effort is equal and the best, and for the time, the *MMRE* of IWD is the smallest. The *MMRE* of GA for effort and time is larger than that of both PDBO-CO and IWD. Prediction (0.25) (*PRED*(0.25)) of PDBO-CO and of GA for effort and time is almost equal and the best. *PRED*(0.25) of IWD for effort and time is smaller than that of PDBO-CO and GA. The *MMRE* and *PRED*(0.25) of COCOMO II PA for effort and time is the worst among all. Requirements (Req), Design (Des), Coding (Code) and total defects obtained by IWD algorithm and the original COQUAMO model are closer to the actual ones. By doing a comparison among PDBO-CO, IWD, GA and the original COQUAMO model regarding prediction accuracy, the *MMRE* and *PRED*(0.10) of IWD, PDBO-CO and GA are somewhat comparable. The *MMRE* and *PRED*(0.10) of original COQUAMO model is the worst, except for Code defects is somewhat better. PT of PDBO-CO is the lowest and PT of GA is the second lowest whereas PT of IWD is the largest.

For the **cost-quality model which doesn't depend on dataset**, we conclude: The results for all goals (Faster Better Cheaper FBC, Better Cheaper BC, Better Faster BF and Cheaper Faster CF) are stable for 10 runs of PDBO and are satisfactory, whereas for IWD and GA are unstable. For the two projects, PDBO results generally are the best, whereas IWD results are the worst. GA results generally are the second best. For all goals, PT is recorded in ascending order as follow: PDBO, GA and IWD.

For solving the **DTCQTP**, we conclude: The results regarding minimum total cost, EVO, GA, PDBO are comparable. ACO is better than IWD in some quality bounds but fails to reach some large quality bounds. The results regarding minimum total time and maximum quality are different; in some quality bounds, one algorithm is the best and in others is the worst. The results regarding success rate (SR) for all bounds, IWD, ACO and PDBO have 100% SR whereas GA and EVO SR are different from quality bound to

another. The results regarding processing time for all quality bounds are recorded in ascending order as follow: GA, ACO, PDBO, EVO and IWD.

For **function optimization**, we conclude: PDBO-CO reached the optimum values of the selected first four functions. For the last function, IWD-CO reaches the optimum value whereas PDBO-CO reached the value that is much near the optimal value. Processing time of PDBO-CO for *the selected five functions* is smaller than that of IWD-CO.

Table of Contents

	Title	Page
	Acknowledgments	V
	Abstract	VI
	Table of Contents	IX
	List of Tables	XII
	List of Figures	XIII
	List of Abbreviations	XVI
	List of Symbols	XX
	List of Publications	XXIX
Chapter 1	Introduction	
1.1	Introduction	1
1.2	Terminologies	6
1.3	Software Development Time, Cost and Quality Preliminaries	6
1.3.1	Software Development Time	6
1.3.2	Software Development Cost	6
1.3.3	Software Quality	8
1.4	Time, Cost and Quality in the System Development Life Cycle (SDLC) Phases	12
1.5	Cost, Schedule and Quality Models Related to Our Study	17
1.5.1	Constructive Cost Model (COCOMO)	17
1.5.1.1	COCOMO Model limitation and Benefits	22
1.5.2	Constructive Quality Model (COQUAMO)	23
1.5.3	COQAULMO Integrated with COCOMO II	27
1.6	Problem Statement	28
1.7	Dissertation Objective	28
1.8	Dissertation Organization	29
Chapter 2	Literature Review	
2.1	Introduction	31
2.2	Meta-Heuristic Algorithms	31
2.2.1	Intelligent Water Drops (IWD) Algorithm	31
2.2.2	Ant Colony Optimization Algorithm (ACO)	34
2.2.3	Genetic Algorithm (GA)	35
2.2.4	Egyptian Vulture Optimization (EVO)	36
2.3	Software Estimation Techniques	38
2.3.1	Software Time, Cost and Quality Estimation Problems	38
2.3.2	Accurate Software Time, Cost and Quality Estimation Benefits	39
2.3.3	Software Time, Cost and Quality Estimation Difficulties	39
2.3.4	Algorithmic Methods	40
2.3.5	Non-Algorithmic Methods	42
2.3.6	Summary of Strength and Weakness of the Mentioned Methods	44
2.4	Software Cost Estimation	46
2.5	Software Quality Estimation	49
2.6	Time, Cost and Quality Trade-off Problem	52

Chapter 3	Time, Cost and Quality Trade-off Problem Solutions	
3.1	Introduction	55
3.2	The Proposed Problem data based optimization (PDBO) Algorithm	55
3.2.1	Problem Data Based Optimization Algorithm for Continuous Optimization Problems (PDBO-CO)	56
3.2.1.1	The Proposed PDBO-CO Algorithm	56
3.3	Cost-Quality Models Using IWD Algorithm	59
3.3.1	An Optimized Cost-Quality Model (Local Dataset)	59
3.3.1.1	COCOMO II PA Model Optimization	60
3.3.1.1.1	Assumptions and Representation	60
3.3.1.1.2	The Proposed IWD Algorithm for COCOMO II PA Model Optimization	61
3.3.1.1.3	The Adapted PDBO-CO Algorithm for COCOMO II PA Model Optimization	64
3.3.1.1.4	GA for COCOMO II PA Model Optimization	64
3.3.1.2	COQUAMO Model Optimization	66
3.3.1.2.1	Assumption and Representation	67
3.3.1.2.2	The Proposed IWD Algorithm for COQUAMO Model Optimization	67
3.3.1.2.3	The Adapted PDBO-CO Algorithm for COQUAMO Model Optimization	72
3.3.1.2.4	The Adapted GA for COQUAMO Model Optimization	73
3.3.2	Cost-Quality Model (No Dataset)	74
3.3.2.1	Assumption and Representation	76
3.2.2.2	IWD Algorithm for Finding The Best Project Options	77
3.2.2.3	PDBO Algorithm for Finding The Best Project Options	79
3.2.2.4	GA for Finding The Best Project Options	82
3.4	Discrete Time-Cost-Quality Trade-off Problem in Software Projects	83
3.4.1	Assumptions about Modes, Task Duration, Cost and Quality	83
3.4.1.1	Task Quality Expressed (Given) by Its Defects	83
3.4.1.2	Task Quality Estimation by COQUAMO Model	84
3.4.2	DTCQTP Representation and Modeling	86
3.4.2.1	DTCQTP Representation and Assumptions	86
3.4.2.2	Mathematical Modeling	86
3.4.3	Intelligent Water Drops Algorithm for DTCQTP	87
3.4.4	Ant Colony Optimization Algorithm for DTCQTP	90
3.4.5	Proposed Problem data based optimization Algorithm for DTCQTP	91
3.4.6	Genetic Algorithm for DTCQTP	93
3.4.7	Egyptian Vulture Optimization for DTCQTP	93
Chapter 4	Experimental Results and Analysis	
4.1	Introduction	95
4.2	Cost-Quality Models	95
4.2.1	The Optimized Cost-Quality Model (Local Dataset)	95
4.2.1.1	COCOMO II PA Model Optimization	96
4.2.1.1.1	Dataset Description	96

4.2.1.1.2	Results Analysis	96
4.2.1.2	COQUAMO Model Optimization	99
4.2.1.2.1	Dataset Description	99
4.2.1.2.2	Results Analysis	102
4.2.2	Cost-Quality Model (No Dataset)	110
4.2.2.1	No Project features for The Project Except Its Size	110
4.2.2.2	Little Project features for The Project	113
4.3	Discrete Time- Cost-Quality Trade-off Problem in Software Projects	116
4.3.1	Example 1	116
4.3.1.1	Task Quality Expressed (Given) by Its Defects	116
4.3.1.2	Task Quality Estimation by COQUAMO Model	121
4.3.2	Example 2	123
4.3.2.1	Parameters Initialization	123
4.3.2.2	Performance measures	124
4.4	Problem Data Based Optimization Algorithm for Continuous Optimization Problems	127
4.4.1	Benchmark Functions	127
Chapter 5	Conclusions and Future Works	
5.1	Conclusions	128
5.1.1	The Optimized Cost-Quality Model Based on Dataset	128
5.1.2	The Cost-Quality Model (No Dataset)	129
5.1.3	The DTCQTP	129
5.1.4	Function Optimization	130
5.2	Future Works	130
References		131
Appendices		
Appendix A	DI-drivers Rating Values of COQUAMO	140
A.1	Initial DI-drivers Rating Values of COQUAMO	140
A.2	DI-drivers Rating Values of COQUAMO Used for NASA Projects	143
Appendix B	Application Program and User Interactions	146
B.1	The Main interface of the Application Program	147
B.2	The Interface of COCOMO II PA and COQUAMO Optimization	147
B.3	The Interface of Optimized Cost-Quality Model (Local Dataset)	150
B.4	The Interface of Cost-Quality Model (No Dataset)	151
B.5	The Interfaces of DTCQTP	152
B.6	The Interface of PDBO-CO for Continuous Functions Optimization	154

List of Tables

Table	Table Name	Page
Table 1.2	Basic COCOMO model	18
Table 1.3	Intermediate COCOMO model	19
Table 1.4	COCOMOII PA Model Effort Multipliers	21
Table 1.5	COCOMOII Scale Factors	21
Table 1.6	Defect Removal Fraction DRF values	26
Table 2.1	Advantages and disadvantages of the existing software cost estimation methods	45
Table 3.1	If then rules	85
Table 4.1	Estimated development Effort values	97
Table 4.2	Estimated development Time values	97
Table 4.3	Performance measure comparison	98
Table 4.4	Defects potentials and DIRs	100
Table 4.5	Estimated development Defects values by current COQUAMO coefficients	103
Table 4.6	Estimated development Defects values by optimized IWD coefficients	104
Table 4.7	Comparing Req, Des, Code and total Defects values vs. size by IWD and COQUAMO	105
Table 4.8	Performance measure comparison (not used DIRs)	107
Table 4.9	Performance measure comparison (used DIRs)	109
Table 4.10	Stability comparison (no project features)	110
Table 4.11	Solution quality comparison (no project features except its size)	111
Table 4.12	Processing time comparison(no project features)	113
Table 4.13	JPL flight software data	114
Table 4.14	Solution quality comparison (little project features)	114
Table 4.15	Execution modes of activities(Example1)	117
Table 4.16	IWD algorithm results (Example1)	117
Table 4.17	PDBO algorithm results (Example1)	118
Table 4.18	IWD and PDBO algorithms comparison(Example1)	119
Table 4.19	Execution modes of activities (Example1) and task quality estimation	122
Table 4.20	PDBO results (Example1, task quality estimation)	122
Table 4.21	Execution modes of activities (Example2)	123
Table 4.22	Comparing among algorithms based on quality of solution	124
Table 4.23	Comparing SR and PT among algorithms(Example 2)	125
Table 4.24	The benchmark functions, which are used for testing the PDBO-CO algorithm	127
Table 4.25	The results of the PDBO-CO and IWD-CO	127
Table A.1	Initial DI-drivers rating values of COQUAMO	140
Table A.2	DI-drivers rating values of COQUAMO used for NASA projects	143

List of Figures

Figure	Figure Name	Page
Figure 1.1	Project success	2
Figure 1.2	The trio of project objectives	2
Figure 1.3	Three main groups of people who care about the software's quality	9
Figure 1.4	Three aspects of software quality: functional, structural, and process quality	10
Figure 1.5	The phases and deliverables of the SDLC in the waterfall model	12
Figure 1.6	Basic COCOMO model	18
Figure 1.7	Intermediate COCOMO model	19
Figure 1.8	COCOMO II Model	20
Figure 1.9	Constructive quality model (COQUAMO).	23
Figure 1.10	The cost/schedule/quality model: COQUALMO integration with COCOMO II	27
Figure 1.11	Estimated residual defect densities using COQUALMO	27
Figure 2.1	Pictorial view of GA steps	35
Figure 2.2	Egyptian Vulture at work	36
Figure 2.3	Steps of Egyptian Vulture optimization algorithm	36
Figure 2.4	Pictorial view of pebble tossing	36
Figure 2.5	Pictorial view of rolling with twigs for $DS = 2$.	37
Figure 2.6	Pictorial view of change of angle.	37
Figure 3.1	Continuous optimization problems representation	57
Figure 3.2	The proposed PDBO-CO algorithm	59
Figure 3.3	The architecture of the optimized cost-quality model (local dataset)	60
Figure 3.4	COCOMO II PA model coefficients representation	60
Figure 3.5	IWD algorithm for COCOMO II PA model optimization (IWD_COCOMO II)	64
Figure 3.6	PDBO-CO algorithm for COCOMO II PA model optimization	65
Figure 3.7	GA for COCOMO II PA model optimization	66
Figure 3.8	COQUALMO model coefficients representation	67
Figure 3.9	IWD algorithm for COQUAMO model optimization (IWD_COQUAMO)	71
Figure 3.10	PDBO-CO algorithm for COQUAMO model optimization	72
Figure 3.11	GA for COQUAMO model optimization	73
Figure 3.12	Architecture of the cost-quality model (no local dataset)	75
Figure 3.13	Factors representation	76
Figure 3.14	The IWD algorithm for finding the best project options	79
Figure 3.15	The PDBO algorithm for finding the best project options	82
Figure 3.16	GA for finding the best project options	83
Figure 3.17	Task quality estimation by COQUAMO model	86
Figure 3.18	DTCQTP representation	86
Figure 3.19	IWD algorithm for DTCQTP	89
Figure 3.20	ACO algorithm for DTCQTP	91
Figure 3.21	PDBO algorithm for DTCQTP	92

Figure 3.22	GA for DTCQTP	93
Figure 3.23	EVO algorithm for DTCQTP	94
Figure 4.1	Comparison among Effort values vs. Size	97
Figure 4.2	Comparison among Time values vs. Size	98
Figure 4.3	Performance measure comparison	99
Figure 4.4	Comparison among Defects values by COQUAMO	104
Figure 4.5	Comparison among Defects values by IWD	105
Figure 4.6	Comparison among Req Defects values by IWD and COQUAMO	106
Figure 4.7	Comparison among Des Defects values by IWD and COQUAMO	106
Figure 4.8	Comparison among Code Defects values by IWD and COQUAMO	106
Figure 4.9	Comparison among Total defects values by IWD and COQUAMO	107
Figure 4.10	Performance measure comparison (not used DIRs)	108
Figure 4.11	Performance measure comparison(used DIRs)	109
Figure 4.12	FBC (no project features)	112
Figure 4.13	BC (no project features)	112
Figure 4.14	BF (no project features)	112
Figure 4.15	CF (no project features)	113
Figure 4.16	Processing time comparison	113
Figure 4.17	FBC (little project features)	115
Figure 4.18	BC (little project features)	115
Figure 4.19	BF (little project features)	115
Figure 4.20	CF (little project features)	116
Figure 4.21	Project network (Example1)	117
Figure 4.22	Time-cost-quality trade-off analysis by IWD algorithm (Example1)	118
Figure 4.23	Processing time taken by IWD algorithm (Example1)	118
Figure 4.24	Time-cost-quality trade-off analysis by PDBO algorithm(Example1)	119
Figure 4.25	Processing time taken by PDBO algorithm (Example1)	119
Figure 4.26	Total quality comparison between IWD and PDBO (Example1)	120
Figure 4.27	Total cost comparison between IWD and PDBO (Example1)	120
Figure 4.28	Total time comparison between IWD and PDBO (Example1)	121
Figure 4.29	Processing time comparison between IWD and PDBO (Example1)	121
Figure 4.30	Time-cost-quality trade-off analysis by PDBO (Example1, task quality estimation)	122
Figure 4.31	Project network (Example2)	123
Figure 4.32	Total quality comparison between IWD and PDBO (Example2)	124
Figure 4.33	Total cost comparison between IWD and PDBO (Example2)	125
Figure 4.34	Total time comparison between IWD and PDBO (Example2)	125
Figure 4.35	SR comparison(Example2)	126
Figure 4.36	PT comparison(Example2)	126
Figure B.1	Application program and user interactions	146
Figure B.2	The main interface	147
Figure B.3	COCOMO II PA and COQUAMO models calibration interface	148
Figure B.4	Results of applying IWD for optimizing COCOMO II PA model	148
Figure B.5	Graph chart (10 project) of applying IWD for optimizing COCOMO II PA model	149
Figure B.6	Results of applying IWD for optimizing COQUAMO model	149
Figure B.7	Graph chart (10 project) of applying IWD for optimizing	150

	COQUAMO model	
Figure B.8	Optimized cost-quality model (based on Dataset) interface	150
Figure B.9	Cost-quality model (No Dataset) interface	151
Figure B.10	PDBO results for FBC goal (No Dataset)	151
Figure B.11	The interface of DTCQTP in software projects	152
Figure B.12	PDBO for solving DTCQTP	152
Figure B.13	DTCQTP application interface	153
Figure B.14	The interface for comparing algorithms for DTCQTP	153
Figure B.15	The interface of testing PDBO-CO for function optimization	154