Developing Method for Optimizing Cost of Software Quality Assurance Based on Regression-Based Model

Vu Dao-Phan(✉), Thang Huynh-Quyet, and Vinh Le-Quoc

School of Information and Communication Technology,
Hanoi University of Science and Technology, Hanoi, Vietnam
dpvu@moet.edu.vn, thanghq@soict.hust.edu.vn, vinhlq199@gmail.com

Abstract. In this paper we present a method for Optimizing Cost of Software Quality Assurance base on Regression-based Model proposed by Omar AlShathry [1,2]. Based on the regression-based model, regression analysis to estimate the number of defects in software, we propose an optimal method for software quality assurance based on the constraint conditions using linear programming techniques. The results of a detailed analysis of the theoretical and empirical models are presented and evaluated.

Keywords: Regression-based Model · Software Quality · Optimizing Cost

1 Introduction

In the software development process, the project manager is always interested in three constraints: cost, schedule and quality since the models above cannot accurately determine the trade-off between the constraints. The software cost estimation models such as COCOMO [4] and COQUALMO [4,5], the software quality process standards such as ISO 9126 [5] used to predict the development effort, defect estimation and quality assessment software will be built. However, models based on data analysis of many previous software projects may encounter difficulties for an organization to adjust the fit of the model. Moreover, these models do not show the balance issues between three software constraints.

Cost of software quality (CoSQ) won the major concern of the project managers because it has been estimated that approximately 40% of the software budget is not reasonably used in the defect discovery and removal process [1]. The regression model provides the project managers with ability to control investment capital to ensure software quality by implementing optimization techniques based on the data manipulation of historical projects [1,2]. In addition, based on the model, the project managers and QA practitioners can handle and deal with unforeseen difficulties related to the software development process [3,4]. It also brings out the best solution for quality assurance decisions for the project managers and QA practitioners to deal with budget shortages, reduced schedule or to achieve goals such as minimum quality cost, successful defect removal [5]. Based on the Regression-based Model [1,2], we present

P.C. Vinh et al. (Eds.): ICTCC 2014, LNICST 144, pp. 243–253, 2015.
DOI: 10.1007/978-3-319-15392-6_24

our approach to develop a method for optimizing the cost of Software Quality Assurance: using classification of software phases into products based on the available risk level; using data storage of quality assurance techniques to store detailed information about the quality assurance activities; using improved matrix containing defects to help accurately determine the efficiency of defect removal of the applied quality assurance techniques. We proposed also to apply the least squares algorithm into linear regression to estimate the number of defects in software. The paper also presents an optimal model which applies linear planning problem to generate optimal solutions for quality assurance plan based on the defined constraint conditions. To build testing software, we studied to install Simplex Algorithm and use LINDO API library [7] to solve the problem in the optimization model.

The content of the paper is presented as follows: Section 2 introduces Regression-based Model in details; Section 3 presents the proposed method; Section 4 provides the results of experimental settings and evaluates the results; Section 5 presents the final conclusion and the development direction of the research.

2 Regression-Based Model

Theoretical regression-based model includes two main components [1,2]: regression analysis and computation to find optimal solutions for quality assurance costs. Regression analysis including 2 processes: data collection and analysis. Figure 1 describes the process of modeling activities. The estimation calculation of the costs as input for Linear programming problem, combined with the known boundary conditions to obtain the output is an optimal solution for software quality assurance plan. Before collecting data, it is necessary to go through the phase classification process of the artifacts into the specific risks. Figure 2 describes the model overview including the phase classification process of the artifacts into the specific risks and quality assurance activities in each phase.

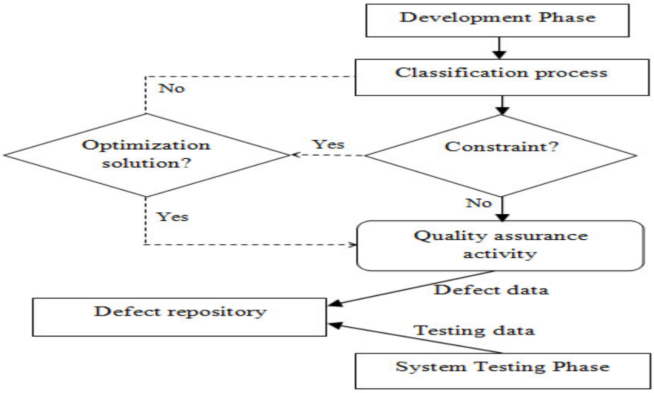


Fig. 1. Work-flow of the Quality Model [9]

Through the two processes mentioned above, the data model will be stored and processed in database, which helps make decisions for projects in the future. To estimate costs, the model bases on the data of the projects in the past but only gets from a single organization, storing and analyzing data generated from the quality assurance activities of the organization. After product sorting process, the quality assurance group stores all the data related to the details of the product. The details include: phase, phase size, type of artifacts, size, and rate of products. In each phase, the input data consist of two interconnected boards: the type of products of each phase and the QA techniques assigned to each product as follows: the cover of each technique, the number of defects found and the number of defects overlooked corresponding to each technique.

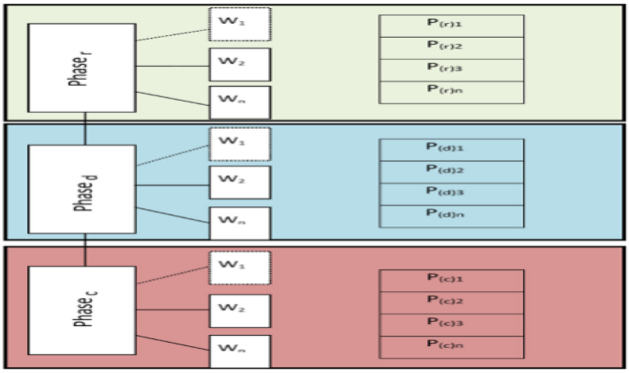


Fig. 2. Model overview [9]

The proposed model that improved defect containment matrix is used to analyze the effectiveness of the technique in order to verify, to validate and to monitor operations and then to detect and remove the defects at each phase. Throughout the development process, the project managers can analyze the effectiveness of quality assurance techniques as a whole for each development phase. Matrix plays an important role in the process of collecting and analyzing data.

$$DRE = \frac{Number\ of\ defects\ found}{Total\ defects} \tag{1}$$

The process of data analysis group of variables associated with each analysis, thereby determining the relationship between them. The analysis principle bases on the average values of the variables and the regression analysis. To build the decision support system based on the variables in the model, it is necessary to determine the relationship between the number of defects in each product and the size of the work product. The relationship can be in two forms: linear relationship and non-linear relationship. Many studies have shown that the growth of software size tends to increase the number of defects [1,2,8,9]. To increase the accuracy of the model, it is supposed that there is always a linear relationship between the size and the total number of defects found. The goal of linear regression analysis for a set of data points is to solve the following equation denoting the best-fit trend line between those data points:

$$y= m*x + b$$

Where: y is a number of defects in a work product; x is a size of work product; m is the slope-intercept between the two variables x, y; b is a constant.

After the regression analysis, we obtain a line graph of the values of two variables (x, y) connecting the number of defects with the size of the software work product:

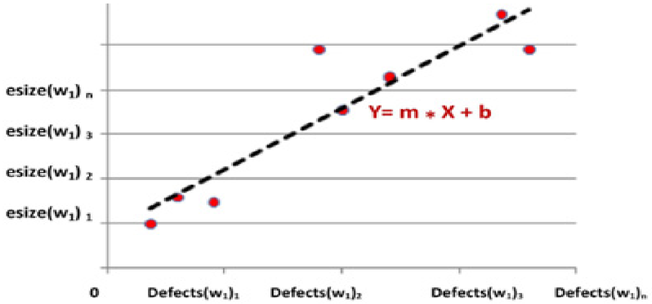


Fig. 3. Proposed Regression Analysis [9]

Since then, the QA team can use the equation as a foundation to predict the total estimated defects in the products of a particular type in a particular phase of the software development lifecycle.

Least squares method to determine the parameters of m and b:

For the data set$\{(x_1, y_1),\ldots,\{(x_n, y_n)\}$,we need to determine the linear equation $y = m*x + b$ so that the expectation *E(m,b)* achieves the smallest value using the formula [9]:

$$E(m,b) = \sum_{n=1}^{N}(y_n - (mx_n + b^2))^2 \qquad (2)$$

For the boundary value, when |*m*| and |*b*| are larger the *E(m,b)*is the greater, it should not need to consider the boundary value.

The goal is to find the values of m and b to obtain the smallest *E(m,b)*.

We calculate the differential for each component of m and b [9]:

$$\frac{\partial \mathrm{E}}{\partial \mathrm{m}} = \sum_{\mathrm{n}=1}^{\mathrm{N}} 2(\mathrm{y_n} - (\mathrm{mx_n} + \mathrm{b})(-\mathrm{x_n}))$$

$$\frac{\partial \mathrm{E}}{\partial \mathrm{b}} = \sum_{\mathrm{n}=1}^{\mathrm{N}} 2(\mathrm{y_n} - (\mathrm{mx_n} + \mathrm{b}))$$

Rewriting the equation:

$$(\textstyle\sum_{n=1}^{N} {x_n}^2) * m + (\sum_{n=1}^{N} x_n) * b = \sum_{n=1}^{N} x_n y_n$$

$$(\textstyle\sum_{n=1}^{N} x_n) * m + (\sum_{n=1}^{N} 1) * b = \sum_{n=1}^{N} y_n$$

The above equations can be expressed as a matrix M:

$$\begin{pmatrix} \sum_{n=1}^{N} {x_n}^2 & \sum_{n=1}^{N} x_n \\ \sum_{n=1}^{N} x_n & \sum_{n=1}^{N} 1 \end{pmatrix} \binom{m}{b} = \begin{pmatrix} \sum_{n=1}^{N} x_n y_n \\ \sum_{n=1}^{N} y_n \end{pmatrix}$$

Calculate the determinant of the matrix M:

$$detM = N^2 * \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})^2 \tag{3}$$

If the different values x_i with $\mathrm{i} = \overline{1, \mathrm{N}}$ then det$M$ is always different from 0. Then it easily calculates the matrix *(m,b)* by multiplying the right-hand side matrix with the inverse matrix of the left-hand side coefficient matrix.

As a result we obtain a formula to calculate the parameters m and b [1,2]:

$$b = \left(\sum_{n=1}^{N} {x_n}^2 * \sum_{n=1}^{N} y_n - \sum_{n=1}^{N} x_n y_n * \sum_{n=1}^{N} x_n \right) / detM \tag{4}$$

$$m = \left(N * \sum_{n=1}^{N} x_n y_n - \sum_{n=1}^{N} x_n * \sum_{n=1}^{N} y_n \right) / detM \tag{5}$$

$$detM = \left(N * \sum_{n=1}^{N} {x_n}^2 - \sum_{n=1}^{N} x_n * \sum_{n=1}^{N} x_n \right) \tag{6}$$

The computational complexity of the algorithm in the worst case is O(n^3).

3 A Proposed Method for Optimizing Cost

3.1 The Optimization Model Structure

The proposed model as the basis for the process of making decisions for QA activities includes three interrelated components: (1) Estimated number of defects is detected and ignored; (2) Cost and time of a QA technical and (3) Cost incurred due to defects overlooked. The number of defects detected by QA p technique is the estimated number of defects found in the product w that depends mainly on the value of the experience derived from the regression analysis process of the past projects and the estimated size of the final product [9]:

$$eD_w = I_w * \text{esize}(w) \tag{7}$$

Among them: Iw is the defect infection rate of each KLOC in product w of phase x; esize(w) is the estimated size of product w. We have the formula to estimate the total number of defects found in phase x [9]:

$$N_x^{found} = \sum_{w \in W} \sum_{p \in P} \beta(p) * eD_w * DRE_p \quad (8) \tag{8}$$

With DRE_p is the value of the defect removal effectiveness of QA p technique; β_p is the coverage of the QA p technique compared with the overall size of the product in a quality assurance activities. Overall condition: $\sum_{p\in P}\beta(p) \leq 100\%$. Total number of defects overlooked in phase x [9]:

$$N_x^{escaped} = \sum\nolimits_{w\in W}\sum\nolimits_{p\in P}\beta(p) * eD_w * (1 - DRE_p) \tag{9}$$

Costs and effort are divided into two parts: the cost to implement a QA technique and costs to remove the defect found. To estimate the time and effort implementing quality assurance techniques, it should use parameters t_p: the average execution time of an application technique QA $p \in P$ for the product $w \in W$. This value is retrieved from the model data source. Unit of measurement is time standard compared to size (hour/FP), and $size_w$: the size of the product.

Total execution time for the entire phase [9]:

$$Ext_x = \sum\nolimits_{w\in W}\sum\nolimits_{p\in P}\beta(p) * size_w * t_p \tag{10}$$

Execution cost is calculated by: $Exc_p = Lr * Ext_p$ (11)

Where: Lr is the coefficient of worker, Ext_p is the amount paid to quality assurer in a unit of time.

The defect removal cost of the technique QAp is:

$$Rc_p = \beta_p * eD_w * \mathrm{DRE}_p * C_p^{removal} * Lr \tag{12}$$

Where: $C_p^{removal}$ is the cost to remove a defect originating from a product $w \in W$ in phase $x \in X$ by technique QA $p \in P$.

Cost arising from defects overlooked by the activities is the cost to eliminate defects overlooked in the development phases and is detected in the test phase:

$$Esc_p = \beta_p * eD_w * (1 - \mathrm{DRE}_p) * C_w^{escaped} * Lr \tag{13}$$

With: $C^{escaped}$ is incurred cost for each defect overlooked.

In some cases of the high-risk products, the project managers can apply a combination of at least two QA techniques simultaneously to reduce infection rate of defects in the next phases.

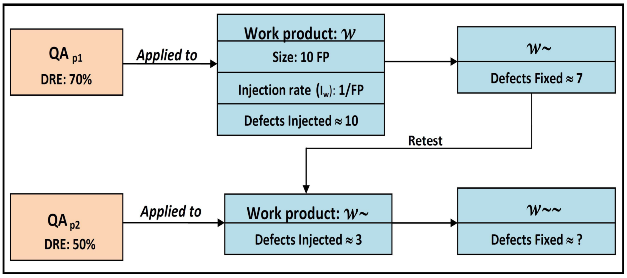


Fig. 4. Combining QA Practices [9]

The variable λ is the probability so that the technique QA p_2 finds the defects which are different from the defects detected by p_1.

The number of defects detected after applying consecutive combination p_1-p_2 [9] is as follows:

$$N_w^{found} = \beta_{p1} * eD_w * \mathrm{DRE}_{p1} + \beta_{p2} * eD_w * (1 - \mathrm{DRE}_{p1}) * \lambda_{p1\text{-}p2} \tag{14}$$

The cost saving from quality assurance activities is the cost of applying one paid QA technique in the testing phase in order to evaluate the cost-effectiveness of two potential QA plans in terms of cost saving in the future compared with the current estimated cost. Cost savings [9]:

$$Sc_p = \beta_p * eD_w * \mathrm{DRE}_p * C^{escaped} \tag{15}$$

The overall cost is:

$$\begin{aligned} \textit{Total development cost} &= \textit{The cost of product development} \\ &+ \textit{The cost of quality} \end{aligned} \tag{16}$$

The cost of product development can be estimated by COCOMO model.

Return on investment is as follows: $ROI = \frac{Value}{Cost}$ (17)

Among them: *value* is the savings cost of fixing defects found in the testing phase; *cost* is equals the effort of both executing the QA practice and fixing defects found.

Return on investment *ROI* of all QA techniques applied to product *w*:

$$ROI_w = \frac{\sum_{p \in P} Scp - (\sum_{p \in P} Scp + Rcp + Esp)}{\sum_{p \in P} (Excp + Rcp + Esp)} \tag{18}$$

3.2 An Optimization Method

The input conditions for the optimization problem in the model consist of three parts: type of optimization, objective function and constraints. Optimization type in the present case, we need to solve the optimization problem with minimum cost, which is equivalent to achieving the desired DRE value with minimum cost. QA costs include 3 categories: execution cost, elimination cost and incurred cost. It is necessary to consider the sides of the QA costs. Therefore, optimization type is to minimize the overall costs generated by QA process undertaken by any QA technique distribution. The objective function is synthesized from the types of costs: execution cost, execution time, execution cost, defect removal cost and incurred cost. From that we can calculate the overall cost function as follows:

$$TotalCost_{x,y,z} = Exc_p(\mathrm{x,y,z}) + Rc_p(\mathrm{x,y,z}) + Esc_p(\mathrm{x,y,z}) \tag{19}$$

The constraints established by the project managers: the defect removal effectiveness value of the desired DRE, the coverage of the QA techniques, execution cost. Linear programming method is used to solve the optimization problem in the model of the form [7,8].

4 Experiments and Evaluation

4.1 The Testing

The model is mainly aimed at medium and big sized projects which often have development period from 8 months to 1 year. The model needs to collect data from over 20 software projects of an organization to develop the data source before being used as a decision support system for quality assurance plans. For testing data of the model, the organizations should have verification and validation activities (V&V).

4.2 Experimental Software Development

The programmed model is simulated by JAVA language in Netbeans IDE and in Windows 7 64 bit operating system environment. The model uses LINDO API providing the means for software developers to integrate optimization into their programs [6,7]. LINDO API is designed to solve many optimization problems including the linear programming problem. LINDO API represents discrete matrix to store the coefficient matrix in the model. It represents the matrix through 3 or 4 vectors.

4.3 The Experimental Results

Typical Case Study of Experiment: Company X applies the regression-based model to manage and control QA activities. After applying the model to some software projects, they were able to develop a significant data source containing QA data of all QA activities of the projects in the past.

Input Data: Data are classified and analyzed to fit the model, to help define the necessary values for the parameters in the model.
Company X launches a new software project and applies the software model for accurate estimation of the expected outputs of the QA plans applied for the project.
Project P with estimated size: 20000 KLOC, development period: 2 months, labor: 3000 persons/ month.

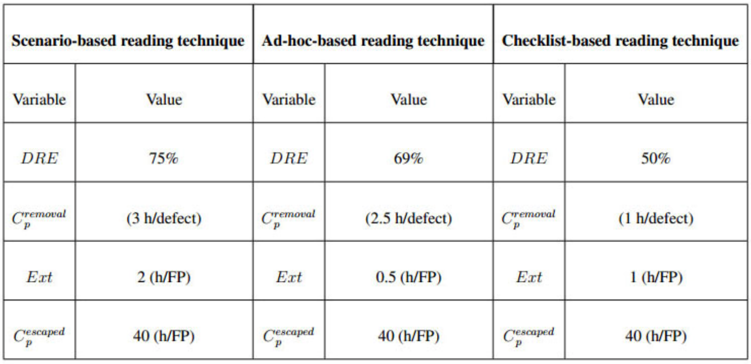
After product sorting process: document making phase has FP 100 product sizes at high-risk level (the important specification).

Applying the regression-based model, infection rate of the high-risk products is predicted, I = 0.4 defects/FP.

Available 3 QA techniques that can be applied in a total of 9 techniques have been applied to the same type of product in the previous projects.

The project managers can set constraints such as the value of defect removal effectiveness (DRE) for the 3 desired QA techniques applied is $\overline{DRE} = 60\%$, and the total coverage of all three techniques is 100%. That means that all products with the specific risks will be checked and maintained the defect removal effectiveness to be 60% with minimal cost.

Table 1. Details of the QA technique applied in the project of Company X



Scenario-based reading technique		Ad-hoc-based reading technique		Checklist-based reading technique	
Variable	Value	Variable	Value	Variable	Value
DRE	75%	DRE	69%	DRE	50%
$C_p^{removal}$	(3 h/defect)	$C_p^{removal}$	(2.5 h/defect)	$C_p^{removal}$	(1 h/defect)
Ext	2 (h/FP)	Ext	0.5 (h/FP)	Ext	1 (h/FP)
$C_p^{escaped}$	40 (h/FP)	$C_p^{escaped}$	40 (h/FP)	$C_p^{escaped}$	40 (h/FP)

Results Using the Optimization Model

X, Y, Z are respectively three techniques: scenario-based reading techniques, ad-hoc reading, checklist-based reading.

With 3 QA techniques above, through linear regression data analysis, we will estimate the number of defects at specific risk level which may occur in project P. Then, to find solutions to ensure software quality with optimizing cost as Linear programming problem, we apply Simplex Optimization Method to obtain the optimization solution:

Find min $(Total.Eff + Total.Esc)_{\mathrm{X,Y,Z}}$ *// Total labor effort + The total overlooked cost*

With assumptions: $\beta_x + \beta_y + \beta_z = 100\%$ $\overline{DRE} = 60\%$

After implementing the Simplex Optimization Method by LINDO API, we obtain the optimization solution with the lowest cost ~ $15,189.47

This cost can be achieved with weights for 3 QA techniques applied respectively:

TechWeightX: $\beta_x = 0\%$ TechWeightY: $\beta_y \approx 53\%$ TechWeightZ: $\beta_z \approx 47\%$

4.4 Evaluation of the Method

Advantages: The regression-based model plays a role as a decision support system combined with the calculation formula to estimate the number of defects infected in the products or the entire development phase of software life cycle. It compares the effectiveness and appropriateness of the different software quality assurance techniques to a specific quality assurance activity in a software organization. Calculating the execution cost and time, defect removal cost of a quality assurance plan. Bringing out the best solution for quality assurance plan based on the three constraints: costs, quality and time. Assessing the quality assurance plan is based on ROI.

Disadvantages: The model only applies linear relation to estimate the number of defects without taking into account the non-linear relation. The linear attribute is only precise if the software development process is stable and the factors that may affect the number of defects are reduced. Functional system is mainly based on the interaction between each development phase of software life cycle and the system testing phase. The process of quantifying values of the model such as defect removal effectiveness, defect increased coefficient, eliminate cost, etc. is based on the links between data generated from quality assurance activities in the development phases and the system testing phase. The model is imprecise due to not taking into account the mutual correlation between the phases of the software life cycle (the previous phases and the next phases). The model should be evaluated based on the actual data from software projects in the same development organization to build a stable and reliable source of data. This process will consume a lot of time.

5 Conclusions and Future Research

We have introduced a regression-based model optimizing cost for software quality assurance using the collected data on the quality assurance activities, studying classification options of software phases into the products based on the available risk level, introducing data storage resources of quality assurance techniques to store detailed information about the quality assurance activities. The improved matrix containing defects will enable us to accurately determine the defect removal effectiveness of quality assurance techniques applied. We also presented theoretical and empirical evaluation of the model. Through the results, it can be stated that the regression-based model optimizing cost for software quality assurance can provide the optimization solution for quality assurance plan based on the defined constraint conditions. The identified result is an optimizing cost value for the quality assurance activities and quality assurance plan accordingly.

However, the model only applies linear relation for estimating the number of defects without taking into account the non-linear relationship, which may be inaccurate due to not taking into account the mutual correlation between the phases of the life cycle, between the previous phases and the next phases. The model should be evaluated based on the actual data from software projects in the same development organization to develop a stable and reliable source of data. This process must consume a lot of time.

Some possible development directions are specifically recommended. Firstly, we should apply the algorithm to handle data, including non-linear relation between the product size and the number of defects detected. Secondly, we study the correlation between the development phases of the software life cycle to increase the accuracy of the model when the quality assurance activities in later development phases can detect and remove defects in the previous phases. Thirdly, we should propose decision support process based on risk: the quality assurance methods are linked to the level of risk associated with them. Each defect can be assigned a found probability value which characterizes the probability of detecting defects in the system testing phase.

References

1. Alshathry, O., Janicke, H.: Optimizing software quality assurance. In: 2010 IEEE 34th Annual on Computer Software and Applications Conference Workshops (COMPSACW), pp. 87–92 (2010)

2. AlShathry, O.: Operational profile modeling as a risk assessment tool for software quality techniques. In: 2014 International Conference on Computational Science and Computational Intelligence (CSCI), vol. 2, pp. 181–184, 10–13 March 2014
3. Lazic, L., Kolasinac, A., Avdic, D.: The software quality economics model for software project optimization. World Scientific and Engineering Academy and Society (WSEAS) **8**(1), January 2009
4. Jones, C.: Estimating Software Costs: Bringing Realism to Estimation, 2nd edn. McGraw-Hill, New York (2007)
5. Alshathry, O., Helge, J., Hussein, Z., Abdulla, A.: Quantitative quality assurance approach. In: NISS 2009 International Conference (2009)
6. Kan, S.: Metrics and Models in Software Quality Engineering, 2nd edn. Addison Wesley (2000)
7. Lindo System Inc., Optimization Modeling with LINGO, 6th edn. Lindo System Inc. (2006)
8. Moore, D., McCabe, G.: Introduction to the Practice of Statistics. W. H. Freeman and Co., London (2003)
9. AlShathry, O.: A Regression-based Model for Optimizing Cost of Software Quality Assurance, De Montfort University (2010)