



Analysis of Optimal Cost Approach for Software Quality Assurance Model

Dr. O. Naga Raju¹, D. Naga Deepa²

¹HOD, Dept of Computer Science Government Degree College Macherla, Guntur (Dt) A.P., India

²Research Scholar Department of Computer Science & Engineering, Acharya Nagarjuna University, A.P., India

Abstract: - *Analysis of software quality assurance cost optimization constitutes a significant part of the total development costs of a software system. Most of the approaches used for quality assurance include fifty percent of defect detection and removal methods. It also points out, how to optimize the investment into various software quality assurance techniques and the expected reliable development of high quality software regularly becomes a major problem. This paper provides one of such approach to optimize the cost involved while maximizing the quality, reliability and also reduces the cycle time for the modeling system.*

Keywords: *Quality Assurance, Optimization, defect detection*

I. INTRODUCTION

Quality assurance for software is a key factor for its success its business value as well as its demand for customers. Quality decides whether the system can be used to acquire more benefits. For a vendor it stands for its efforts for fault removal and addition of new functionalities. Software quality assurance comprises of the methods and techniques which are meant to assure that software has a certain or desired quality .There are basically two approaches for this assurance as (i) Constructive (ii) Analytical. The first one meant to improve the process of software development and to prevent defects and other problems from being introduced, the later one aims to detect the defects and other problems. Defect-detection deals with existing defects whereas defects should not come into existence using defect prevention. Hence, the analysis of these types of techniques should be done separately. Many of the estimates says that quality assurance constitutes about 50 % of the total development cost, yet there is a huge opportunity of saving the cost using analytical techniques.

To save cost for quality assurance, it is need to be optimized in its usage and in the development process with the main to reduce the costs and increase benefits. There are mainly two approaches to fulfill this aim (i) Improving the existing techniques (ii) Use the existing techniques in an optimal cost way. When a technique is to be selected it needs to be determined whether it is applicable to a specific kind of software and its defect detection capabilities. Hence, an approach that helps to distribute the optimal effort calculated using the cost model over the components would be helpful.

II. LITERATURE SURVEY

Many estimates say that analytical quality assurance constitutes about 50 % of the total development cost; this figure is attributed to Myers [1] , Jones [2] still assigns 30-40 % of the development costs to quality assurance and defect removal . A study analysis, National institute of standards and technology of United States [3] even 80 % of the development costs are assigned to the detection and removal of defects. Ntafos discusses in [4] that “cost is clearly a central factor in any realistic comparison but it is hard to measure, data are not easy to obtain, and little has been done to deal with it.” Costs are the only possibility to compare all the influencing factors in SQA because they are the unit all those factors can be reduced to. Because of that Rai et al. identify in [5] mathematical models of the economics of software quality assurance as an important research area. “A better understanding of the costs and benefits of SQA and improvements to existing quantitative models should be useful to decision-makers.”

III. BACKGROUND

Software quality

Quality is neither visible nor tangible and it is immeasurable. This is a physical feeling of the user which causes happiness or comfort with respect to a particular aspect. For instance if a user experience a trouble free manipulation of the software ,then the quality aspect usability of that software is privileged.

There is no particular definition for software quality [6] .It depends upon the satisfaction on the application under certain situation. There are many definition regarding the software quality, Crossby [7] explains his perspective on quality as a conformance to requirements. Walter Edwards [8] states that quality must be defined in terms of customer satisfaction. Armand villain explains his perspective as “Quality is a measure determination, not an engineer’s determination not a marketing determination nor a general determination. Many authors and people have given their own definition and one of them has to do with the quality as an objective reality independent of the existing man.

Quality Approaches

Quality is a difficult concept in itself; Garvin gives in [9] a comparison of different approaches to define quality

- (a) Product Based Approach: Quality describes differences in the quantity of some desired attributes. Hence, in contrast to the transcendent approach, this is precisely measurable
- (b) User based approach: It assumes that the product that satisfies the needs of the user best has the highest quality. This is related to the product-based approach as the desired attributes must be defined by users
- (c) Manufacturing approach: Takes a more internal view and defines quality as conformance with specified requirements. This definition, however, includes all the problems we have with developing exhaustive and useful specifications. It assumes that it is always possible to define the requirements of a product completely and hence a deviation of the specification can be easily recognized
- (d) Value based approach: It assigns costs to conformance and nonconformance and hence can calculate the value of a product. It assumes that we are able to assign a value to all involved factors.

Quality factors

A quality factor can be a feature or characteristic that affects an item's quality. There are different factors which describe about the quality of a software

- ❖ Functionality
- ❖ Reliability
- ❖ Usability
- ❖ Efficiency
- ❖ Maintainability
- ❖ Portability

IV. SYSTEM MODEL

The proposed system model is divided into three components and are depend on spend factor (t) as global parameter. It involves the following factors

(i) **Direct Costs (d(t))**: These costs are those that can be directly measured from the application of a defect detection technique. They are dependent on length (t) of the direct costs for an application (A)

This contains two main blocks setup costs and execution costs; it is dependent on the spent effort A denoted as t_A . The execution costs we can derive the difficulty of detecting the defaults in the software which represents the probability that fault is not detected. If the fault is detected it incurs costs for its removal, the expected values of the costs is given as

$$E\{d_A(t_A)\} = u_A + e_A(t_A) + \sum_i (1 - \theta_A(i, t_A)) V_A(i) \quad (1)$$

Where u_A are setup costs, $e_A(t_A)$ the execution costs, and $V_A(i)$ the fault removal costs specific to that technique

Future Costs: The future costs denoted $E\{f_A(t_A)\}$. It is divided into two parts, fault removal costs $V_F(i)$ and failure effect costs $C_F(i)$

$$E\{f_A(t_A)\} = \sum_i \pi_i (\theta_A(i, t_A)) V_F(i) + C_F(i) \quad (2)$$

Where π_i represents the probability

Revenues: It is considering not only the costs of the defect detection technique but also their revenues. They are essentially saved future costs. We denote the revenues with $E\{r_A(t_A)\}$

$$E\{r_A(t_A)\} = \sum_i \pi_i (1 - \theta_A(i, t_A)) V_F(i) + C_F(i) \quad (3)$$

Based on the three components of the model, we are able to calculate several different economical metric of the quality assurance process. There are metrics total cost, profit, and return on investment. All these metrics can be used for two purposes: (1) an up-front evolution of the quality assurance plan as the expected total cost, profit, or return on investment, and (2) a single post-evolution of the quality assurance of the project.

Total Cost: The total cost describes the sum of all economic costs for producing products. It is one possible metric that can be optimized. Total cost can be calculated by adding the direct costs and the future costs. The expected value of the direct costs dx and future costs fx of the sequence of defect-detection technique applications X.

$$\text{Total Cost} = dx + fx \quad (4)$$

Profit: We describe the gain provided by the quality assurance with the term profit. Hence, it is the revenues less the total cost. It is defined using the three components as: direct costs, future costs, revenues. The expected value of the revenues rx of the sequence of defect-detection technique applications X.

$$\text{Profit} = rx - dx - fx \quad (5)$$

ROI: Another metric used in economic analyses is the return on investment (ROI) of the defect-detection techniques. The ROI- also called rate of return - is commonly defined as the gain divided by the used capital. We use Boehm et al. [10] equation for (Benefits - Costs) / Costs, to calculate the total return on investment (ROI).

$$ROI = \frac{rx - dx - fx}{dx + fx} \quad (6)$$

V. EXAMPLE BASED MATHEMATICAL ANALYSIS

In this analysis the order processing system of mountain bike manufacturer is considered the intention is to determine the ROI of quality investment for the system. In case of unavailability of the order-processing system, the mountain bike manufacturer will not be able to sell mountain bikes during the downtime.

The ROI (rate of Interest) Calculation is as follows:

The rating scale of the mountain bike system includes the Nominal, High , Very High and Extra High ratings from which the availability can be given as

$$\text{Availability} = \frac{\text{mean time between failure}}{\text{mean time between failure} + \text{mean time to repair}} \quad (7)$$

is determined, and the mean time between failure and the mean time to repair is estimated by business domain experts.

Project	Rating	MTBF (Hrs)	MTTR (Hrs)	Availability	Loss (M\$)	Increased Value (M\$)	Cost (M\$)	Change (M\$)	ROI
Mountain Bike	Nominal	300	3	0.9901	5.31	0	3.45	0	-
	High	10k	3	0.9997	0.16	5.15	3.79	0.344	14.0
	Very High	300k	3	0.9999	0.005	0.155	4.34	0.55	-0.72
	Extra high	1M	3	1	0	0.005	5.38	1.04	-1

The results of this analysis initially were discussed with business experts of the mountain bike order-processing quality assurance team who pointed out that a negative ROI that resulted from the improvement of a High to a Very High RELY level would not let their interest in improvements disappear. As when the needs in availability are fulfilled, there emerge other important motivators.

Reducing security risks is an example for a possible motivator.

VI. CONCLUSION

There are the numbers of techniques to verify the cost effectiveness of quality assurance. Cost optimal use analytical quality assurance but we do not distribute the effort between different techniques but we analysis how the effort is best distributed over the components of the system. This is done by identifying the most fault- and failure-prone components based on a metrics suite and detailed design models. The approaches that exist for models either have slightly different aims, analysis dependability attributes or readability, or concentrate on the static structure, analysis the fault-proneness. During the development and quality assurance, we use estimating quality models that assess the current state of the product and process. The example analysis stated in this article gives a clear idea about the optimal estimation of cost and reduction of the resources that are involved

REFERENCES

- [1] Glenford J. Myers, "The Art of Software Testing", John Wiley & Sons, 1979
- [2] Capers Jones, "Applied Software Measurement: Assuring Productivity and Quality," McGraw-Hill, 1991.
- [3] RTI. "The Economic Impacts of Inadequate Infrastructure for Software Testing Planning" Report 02-3, National Institute of Standards & Technology, 2002.
- [4] Simeon C. Ntafos, "On Comparisons of Random, Partition, and Proportional Partition Testing. IEEE Transactions on Software Engineering, 27(10):949-960, 2001.
- [5] Arun Rai, Haidong Song, and Marvin Troutt. Software Quality Assurance: An Analytical Survey and Research Prioritization. Journal of Systems and Software, 40:67-83, 1998.
- [6] Hoyer, R. W. and Hoyer, B. B. Y., "What is quality?" Quality Progress, no. 7, pp. 52-62, 2001
- [7] Crosby P.B., "Quality is Free the Art of Making Quality Certain NY, McGraw -Hill-1979
- [8] Deming & Edwards W, "Out of Crisis", MIT Press-1986
- [9] David A. Garvin. What Does "Product Quality" Really Mean? MIT Sloan Management Review, 26(1):25-43, 1984.
- [10] Barry Boehm, LiGuo Huang, Apurva Jain, and Ray Madachy, "The ROI of software Dependability: The iDAVE model", IEEE software, 21(3): 54-61, 2004.