8-25-1995

Cost of Quality in Software Products: An Empirical Analysis

Mayuram S. Krishnan
Carnegie Mellon University

Sunder Kekre
Carnegie Mellon University

Charles H. Kriebel
Carnegie Mellon University

Tridas Mukhopadhyay
Carnegie Mellon University

Cost of Quality in Software Products: An Empirical Analysis

Mayuram S. Krishnan*
Sunder Kekre*
Charles H. Kriebel*
Tridas Mukhopadhyay*
Marc Kellner**

* Graduate School of Industrial Administration
** Software Engineering Institute
Carnegie Mellon University,
Pittsburgh, PA 15213

1. Introduction

Computer software has emerged as a major worldwide industry, estimated at $450B for 1995 of which $225B is attributable to US firms [Boehm, 1987]. Yet, in many organizations, costs and schedules for software projects are largely unpredictable, and product quality is often poor [DeMarco and Lister, 1993]. This underscores the need to study both the quality of the software product and the life-cycle cost incurred in the development and maintenance of the products. Increasing expenditure in software has caught the attention of researchers. Identifying software productivity factors and estimating software costs continue to be important research topics [Mukhopadhyay and Kekre, 1992; Banker et al., 1993]. Researchers have adopted both empirical and theoretical approaches to better understand the process of software development and maintenance. Though software cost continues to be an important research question, competition in the software industry and the increased role of software in everyday life have also made development cycle time and quality important research issues.

The quality of software has been studied mainly from defect analysis and software maintenance perspectives. Empirical research has analyzed tradeoffs between software quality and maintenance, and examined drivers of software maintenance costs [Banker, et al., 1993]. Practitioners in the software industry are still faced with the challenge of understanding the key tradeoffs in a software project in order to deliver quality products to customers on time and without cost overruns. This underscores the need to study the various factors that influence the life-cycle cost and quality in software products. Moreover, the effect of the process used in a software project on the outcome of the project in terms of cost to the software developer and quality of the product has not been examined rigorously. Thus in this research, we propose to model the life-cycle cost and quality of software products based on the factors related to product, people, process and technology deployed in the software projects.

2. Research Issues

One focus of past research in the area of software product management has been the estimation of cost incurred in developing the product. Reliability assessment of the software products has also received the attention of researchers. However, for software products, life-cycle cost and quality are interrelated. Our notion of the quality of a software product is based on the number of unique and valid customer reported problems. Thus, the maintenance cost of fixing customer reported problems may be substantially higher when the quality of the product is lower. The interrelation between cost and quality of products in manufacturing industries has been addressed in the operations management literature. For example, one view of quality provided by Crosby, claims substantial reduction in the life-cycle cost of the product for increase in quality [Crosby, 1979]. In the case of software products this relationship has not been examined. Moreover, in software development and maintenance, life-cycle cost and quality are jointly determined based on various factors related to product, people and the development process adopted. We next discuss these factors more precisely.

The process of commercial software development often starts with an analysis of the market needs. Consequently a high level design team commences the assessment of a feasible software product for the specific needs of the market. A detailed design work on the product specifications then provides an estimate on some of the product factors such as size and complexity. These product factors may differ significantly based on the type of the software product (e.g., language compilers or application products). The detailed design work is then followed by the implementation of the product in terms of coding and testing. Moreover, technical capabilities of the software professionals working in the project and the selection of appropriate tools and technology for design and development is believed to significantly interact the software development process and impact the results of a software project.

The methods and practices adopted by software development professionals to develop and maintain the software are believed to have significant impact on the project efficiency in terms of development cost, product quality and cycle time of the product. The rationale behind this belief is that disciplined methods and practices are expected to lead to better control over the software development process. In our model, software process related factors are derived from several software quality standards including the ISO-9000-3 standards for software development and the Capability Maturity Model developed at the Software Engineering Institute of Carnegie Mellon University. Development process related factors in our model address important software development process areas such as requirements analysis, definition of the software process, software quality assurance, configuration management, software project tracking and product engineering. The significance of adopting a well defined and standard process in developing and supporting the software products has been emphasized in the research literature and various corporate quality certification programs [Paulk et al., 1993]. It is an important research question to empirically validate the effects of these software process factors on the results of a software project. An issue for further investigation would be the degree to which these process factors differ in their importance across various products.

3. Model

Life-cycle cost of software products is composed of two main components, the development cost and the cost incurred in supporting the product. The development cost is determined by various characteristics of the software product such as size and complexity, development process factors such as requirements management, project planning and product engineering, factors related to the capability of the project team, and the technology adopted in the project. The cost incurred in supporting the product in the field is influenced by the quality of the product delivered and the factors related to the software maintenance process. In our model, life-cycle cost of the software is specified as a function of product quality, and factors related to product, people, technology and development process characteristics. The quality of the software product is determined significantly by the disciplined methods and practices adopted in developing the software, the complexity of the product, and the capability of the software development team. Thus, in our model, delivered quality of the product is specified as a function of development cost, and factors related to product, people, technology and development process characteristics.

4. Research Site and Data Collection

Our research site is one of the large software development laboratories of a Fortune 100 company. This laboratory develops systems for various applications ranging from language compilers and database systems to network communications. The system development life cycle model used for these products was the traditional waterfall model. In the recent past, considerable resources are being deployed at the design and planning stages and also in improving the process adopted for developing the software products. Our study was initiated to assess the effectiveness of these programs.

Our data collection includes gathering information on the cost and quality of a cross section of recently developed products. Our data also includes the following factors which affect the results of a software project: a) product factors such as size of the product and complexity; b) people factors such as capability and experience; c) process factors related to the process adopted in developing the software. In our analysis, these process factors are based on the Software Engineering Institute's Capability Maturity Model (CMM) [Paulk et al., 1993]. Our cost data (in dollar figures) include the costs incurred in all stages of product development and support starting from product planning through service support in the field. All the products in our sample were developed in the same laboratory and only two programming languages were primarily used across these products.

5. Research Status

We have obtained cost and quality for more than 60 software products from one of the leading software development companies. Our data sets also include the size and complexity of the software products. We are in the process of collecting data on the various factors related to software process and project team. We use the standard process

assessment techniques and instruments of the Software Engineering Institute to measure the process factors. We hope to complete our data collection over the summer and present some preliminary results at the conference. We believe that some of the important contributions of this analysis would emerge from joint specification of cost and quality, and the effect of software development process factors.

6. References

1. Banker R.D., S. Datar, C.F. Kemerer, and D. Zweig, "Software complexity and software maintenance costs," Communication of the ACM, 36(11), pp 81-93, 1993.

2. Boehm, B.W., "Improving Software Productivity," IEEE Computer, September, pp 43-50, 1987.

3. Crosby, P.B., Quality is Free: The art of making Quality certain, McGraw-Hill, New York, 1979.

4. Demarco, T. and T.R. Lister, "Software State-of-the-Art," Dorset House Publishing, New York, N.Y., 1993.

5. Mukhopadhyay, T. and S. Kekre, "Software effort models for early estimation of process control applications," IEEE Transactions on Software Engineering, vol 18, no. 10, 1992.

6. Paulk, M , "Software capability evaluation," ver 2.0, Technical report CMU/SEI-94-TR-06, Carnegie Mellon University, Software Engineering Institute, 1994.