Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Performance Evaluation of Software Quality Model

Mrs Neha Agrawal[1], Prof. (Dr.) PRAVEEN DHYANI[2]
[1]Banasthali University
[2]Executive Director, Jaipur Campus, Banasthali University

I. INTRODUCTION

With the advent of Internet revolution and the emergence of knowledge based systems, Quality acquires a wider and more challenging dimension. Quality has evolved and undergone transformation from the inspection era to the quality control regime and then to quality management and finally to the present TQM approach. At every stage of the transformation “Quality” has been attaining wider dimension with respect to Customer focus, continual improvement and has been evolving for addressing increasing demands of customers with respect to delivery of products and services. Quality evaluation in software industry has been a debatable issue with various models emerging to measure the product or service quality. Quality empowered business requires in depth understanding of business goals and objectives. For any industry, whether manufacturing or software invention, it is seen that the measurements of quality are driven by business needs which in turn are driven by market or customer needs. Organizational objectives hold a key factor for measuring quality. The integrated Business system has shareholders, employees, customer and supplier in any value chain. Value chain fundamentals would be the primary focus and value chain components primarily decide the organizational business goals and objectives. A good measurement system for performance evaluation cannot be empirically drawn from mathematical model but by solid value chain understanding, assessment of needs of customers/end–users, shareholders, suppliers and bottom-line identification.

II. QUESTIONNAIRE

The Questionnaire was administered to three companies at various levels in software value chain. The questionnaire captures the metrics shortlisted for proposed model and help in collecting the data from the candidate companies this was an action based research and based one convenience sampling methodology in qualitative research.

One candidate company from each level of software value chain was taken based on access to company’s support, data and state holder response. Alpha technologies were selected to represent software product development a company, beta technologies was selected in IT services and solution space. Gamma technology was selected in the ITES/BPO/KPO space. The researcher had GOOD REFERENCE from these Organizations and thus response time was less in collecting data.

Literature survey: Various concepts of quality, award models existing hues been studied in depth to understand the factors affecting quality. Industry visits, study of various journals provide source for this phase and gaps in the survey resulted in proposing OQM model.

Development of OQM Questionnaire: A questionnaire incorporating new factors researched is developed in the value chain. Refer OQM Questionnaire in appendix A where questions on tier-1 OQM Model has been administrated to Level-1, Level-2 and Level-3 organizations.

Administration of questionnaire: The questionnaire is administered to a group of software companies operating in the value chain through electronic means.

Collect Tier-1 OQM Metrics six metrics Shortlisted which forms the tier-1 OQM framework is collected for three software organizations.

Validation of Model: The model is validated for its application and interpretation in tern of usage and review.

Learning and Consolidation: Learning from OQM Model offers its strengths and weakness and also provides insights into the future opportunities, which could be leveraged using the Model.

III. OBJECTIVES OF RESEARCH

The research aims to propose a software quality model framework which would identify metrics based on strategic objectives are critical success factors (CSF) unlike other models which prescribe metrics. The model would therefore used by all Level-1, Level-2 and Level-3 software (referred to as L1, L2 and L3 respectively) in the proposed software value chain.

The objectives of the research are:

1. Review the status of research in quality concepts and applications.
2. Compare manufacturing and service industry software value chain in terms of measurement.
3. Propose a suitable set of quality measures for different types of software organization.
4. Suggest a framework for developing the quality measurement strategy for a typical software company.
5. Analyze the drawbacks of existing interpretation of software quality as discussed in the introductory session and propose a model which addresses quality issues and concerns expressed in the introductory section of the research work.

IV. METHODOLOGY

Methodology followed for the present research work was divided into five stages as follows:

Stage 1: Problem Formulation

Stage 2: Research Design

Stage 3: Model Development/Design of templates for data collection

Stage 4: Validation of model

Stage 5: Report writing

These stages are described below:

Stage1: Carrying out literature survey and identifying gaps in the literature formulated the research problem. The problem was a need for a model that would focus on enterprise objectives and provide a metric set to achieve critical success factors for various software organizations in value change.

Stage 2: Research design is the sampled to be identified and in the current research world, scope was on Indian IT software organizations and a sampled company from value change was taken. The value change for software companies operating in India was conceived as a part of the research design.

Stage 3: Once the gap was identified from literature survey, a model was proposed which would pick-plug metric based on enterprise objective. However, critical success factors were taken from software organizations operating in the software value change and six metrics were Shortlisted, which formed the model framework. Various templates were designed and a questionnaire was designed to collect data and templates were populated from responses received from three candidate companies.

Stage 4: Discussing its strengths and its weakness and its application to various software companies in the value chain validated Model.

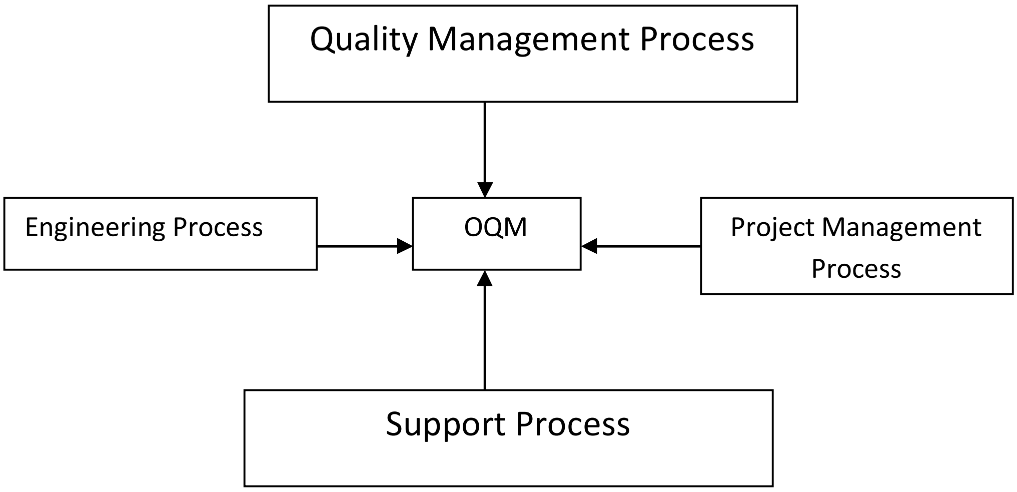
Stage 5: In this stage, the thesis writing is carried out where all the results of the research work are compiled in the form of reports. Chapter scheme is decide where few chapters are devoted for the activity of literature survey, few are devoted in explaining the new model and few are devoted in discussing the limitations and scope of future work. Literature survey would be carried out through published articles, journals and surfing Internet. Analysis was done about the quality management system from the industry visits and in-depth study of various journals.

V. APPLICATION OF OQM MODEL

In the current research work, methods like Case study is used for understanding of complex issue like Software Quality in this research previous research. Case study method details contextual analysis of a limited number of researcher and in general Social scientists has used the Qualitative research method as it is used in the development of OQM Model. In Qualitative research wide use of experience, real-life situations form the basis for application of ideas.

For the development of OQM Model the researcher had adopted the above steps of Case based method. Data was collected for Tier-1 OQM Model and evaluation and analysis was done and the scheme for collection of data for multi-tier was also proposed. The companies were selected based on convenience sampling. The researcher selected the Gamma technologies, Alpha and Beta Technologies close to his place and study. This ensured more interaction, observation and facilitated more idea engineering and brainstorming from the respondents during data collection phase by the researcher This chapter discusses how the proposed OQM Model is a process driven model as it draws data and information along various tiers of BSC at operational, tactical and strategic level in an organization. The chapter focuses on creating a Multi-Tier OQM metric set by an iterative process of using various tools and databases along the value chain.

On the lines of manufacturing value chain propounded by Poter (1980) attempt has been made in the research work to map software value chain with primary activities and secondary activities. Engineering Process like requirement Management, Design, coding, Unit Testing and User Acceptance testing as per SDLC Models addresses primary activities. To ensure the software is of good quality and delivered on time, cost, plan and resources feeder processes like quality Management Processes, Project Management Process is considered as engineering activities.



OQM: A Process Driven Model.

Data Collection: Improvised Third generation Metrics Collection

- First generation metric collection was a manual Personal Software Process (PSP) where multiple forms were filled by individual developers for recording size, defects and effort information. Collection overhead was high, analysis overhead was high and there was high degree of context switching for developers. Third generation metric collection used Hack stat architecture, which has client side sensors, attached to development tools.
- OQM Data Collection advocates usage of Third Generation approach with better control on privacy and sensor availability. OQM model would upgrade the Hack stat architecture of Online Metrics used by proving additional security features and address Generation 4 Personal software Process (PSP). PSP was improvised from manual mode to automation mode in second generation of approaches to metrics collection.

- Stepwise Approach for Data Collection in OQM Model
- Step 1: Data collected at individual level from GUI is picked form emailed Uniform Resource Locators (URL) s) with the help of development tools like DDTS, Clear Case, Junit, CTIM, Project Management Tools and other CASE tools.
- Step 2: Data collected are sensed by sensors, which are tool specific and metric specific.
- Step 3: Web Servers collect data from Sensors and performs analysis and informs users if the data values are outside the Control limits.
- Step 4: Data base keeps tracks of all the data and metrics.
- Step 5: The privacy of the data is secured through Secure Socket Layer (SSL) and through Cryptographic techniques.

VI. Limitations and Scope for future Work

Metrics and Data are significant entities and can help to enhance the other. The coming together of intuition and data can do something more significant. Data refreshes human consciousness. In the long run, data feeds intuition and shapes our minds. Intuition gives data a sense of direction, tenders context and supports data interpretation. The question is not how many metrics we use, but how we use them. Timeliness of metric enhances the chance of metrics entry into decision-making. OQM score card does not address the financial quadrant of BSC in the metric set as the software companies surveyed across three levels did not provide data for financial metric like market share, profitability, ROI, the software value chain. There could be more categories and levels of software companies, which does not from the part, and scope of research work.

When metrics data does not keep pace with organizational informational needs, it indicates a problem. The problem could be too many metrics poor definition of metrics, measurement noise and lack of motivation, inefficient analysis and procrastination in report generation, ineffective communication and poor automation. Metrics Synchronization is a limitation of the research coupled with determination of metric intelligence. Metric Intelligence discusses about the basic intellectual need. Data mining algorithms are not addressed here. The next is the challenge in collecting Software Engineering Data. The challenge of collecting software engineering data is to make sure that the collected data can provide useful information for project, process, and quality management and, at the same time, that the data collection process will not be a burden on development teams. Therefore, the goals of the data collection should be established and the questions of interest should be defined before any data is collected. Data classification Schemes to be used and the level of precision must be carefully specified. The collection form or template and data fields should be protested. The amount of data to be collected and the number of metrics to be used need not be overwhelming. It is more important that the information extracted from the data be focused, accurate, and useful than that it be plentiful. Without being metrics driven, over-collection of data could be wasteful. Over collection of data is quite common when people start to measure software without a priory specification of purpose, objectives, profound versus trivial issues, and metrics and models.

The actual collection process can take several basic formats such as reporting forms, interviews, and automatic collection using the computer system. For data collection to be efficient and effective, it should be merged with the configuration management or change control system. This is the case in most large development organizations. For example, at IBM Rochester the change control system covers the entire development process, and online tools are used for plan change control, development items and changes, integration, and change control after integration (defect fixes). The tools capture data pertinent to schedule, resource, and project status, as well as quality indicators. In general, change control is more prevalent after the code is integrated. This is one of the reasons that in May organizations defect data are usually available for the testing phases but not for the design and coding phases.

The need for online application of metric is felt as a futuristic work in this study. Metrics data is more customarily viewed as historical data and brings benefits from off line analysis. One needs to take metrics from the world of history to reality. The new dimension emerges in the use of metric database when it is fed by rich information environment. The metrics database becomes a large storehouse of assorted data pouring in from various sources. The data contains intelligence,

VII. Learning from Research Work

The use of appropriate metrics by an organization is a movement towards truth. We implement metrics because we wish to deal with true values, we wish to see true picture and we wish to arrive at true solutions. There is no universal method when it comes to implementing metrics. Every organization must build its own method. The whole thing begins with a desire of having a new culture, which accepts transparency of statistical thinking. The tree of metrics taxonomy can really branch into an intricate network covering deep fathoms of software engineering on one hand and penetrating process layers on the other. The central point of implementation is effective use of metrics. Metrics are mirrors that reflect realities. Fear of exposure inhibits the mind from accepting metrics. When the mind becomes ready to see change and reality effective use of metrics is possible.

VIII. Concluding Remarks

OQM Model is a flexible model as it provides options to the project manager and quality manager to pick and choose metrics from the metric set consisting of 47 metrics. The metrics are derived from 8 Goals identified in Software Development Process. Metrics can be iteratively be generated by asking different set of questions each time and thus arriving at multiple Poly morphed OQM Structure. One of the structures discussed in the research work is the Tier-1 where six metrics have been shortlisted. This is one form of short listing where the candidate organizations are taken from three levels in the software value chain. The proposed OQM Model provides the desired flexibility for all levels of Software companies (all the three categories) to pick and choose from the repository of Metric pool (All forty seven Metrics identified initially). Based on the organizational objectives and its affinity to Metric, the Metric is short-listed and data collection is carried out based on the OQM Questionnaire. The key metric governing each of the levels or categories in industry is identified. OQM Model is suited where Software Quality needs to be measured on a long-term basis. It may be noted that, analytic Hierarchy Process (AHP) technique can be used for filtering of metrics while evolving the OQM Model. This is a decision-making technique for rational evaluation of pros and cons concerning different alternative solutions to a multi goal problem. AHP is based on pair wise comparisons and then these comparisons are checked for internal consistency. This method can be adopted while stakeholders can be asked to give their preferences of objectives to measure for a given software company. Based on the relative weights and a predefine cut-off value, a priority vector indicating various objectives can be used for screening purposes. Though this method was not followed in the present research work, a Simple Additive Weights (SAW) method was followed where aggregation was carried out by say C attributes in a cluster and the attributes are research work weights are assigned between Goals and Enterprise Objectives and weights are normalized. Similarly in Level-2 Screening of metrics one can see weights assigned to intermediate metric set AHP is a MCDM (Multi Criteria Decision Making) method. MCDM aims at supporting decision makers with making numerous and conflicting evaluations and highlighting these conflicts and deriving at a compromising solution. This is the strategy that was used in the OQM Model where the organization has multiple objectives to follow and the most appropriate objectives can be worked out. explains the filtering methodology and implicit use of AHP methods like MCDM. Existing method of screening adopts two levels of screening to arrive at Tier-1 OQM Model. At level-1 screening the focus is on taking 8 goals of GQM model and using 47 metrics, which is mapped with 6 enterprise objectives. In OQM model too, Objectives are the first entities to be considered. Both the models focus on Business Critical Success factors first. Measures are arrived at in BSC model after the targets and initiatives are identified. Targets and Initiatives are mapped into Measures in OQM Model. OQM Model is derived from BSC Model. Balanced IT Score Card (BITS) proposed by European Software Institute that provides new version of the four perspectives adding a fifth one i.e. people perspective.

Bibliography

1. Abreu, FB., “Metrics for Object-Oriented Environment,” Proceedings of the Third International Conference on Software Quality, Lake Tahoe, Nevada, 1993, pp. 67-75.
2. Agile Alliance At http://agilealliancebeta.org/article/file /904/file.pdf, 2001
3. Akao, Y., “New Product Development and Quality Assurance system of QFD Standardization and Quality Control”, Japan Standards Association, Vol 25(4) 1997, pp 9-14.

4. Albrecht, A. J., "Measuring Application Development Productivity," Proceedings of the Joint IBM/SHARE/GUIDE Application Development Symposium", 1987, pp. 83-99
5. Development Effort Prediction: A Software Science Validation", IEEE Transactions of Software. Engineering Vol. 9 (6), 1983, pp. 639-648.
6. Arnold, R, S. "A Road Map Guide to Software Reengineering Technology", Software Reengineering. IEEE Computer Society, 1992, pp. 3-22.
7. Azuma, M., "Software Quality Assurance", Vortrags manuscript zum Vortrag, Vol 12(6), Zurich, Germany, 1987.
8. Bailey, C.T., and Dingee, W.L., "A Software Study Using Halstead Metrics", Bell Laboratories Denver, CO. 80234, 1981.
9. Baker, A.L., Beman, J.M., Gustafson, D.A., Melton, A., and Whitty, R.A., "Modeling and Measuring the Software Development Process", Proceedings of the Twentieth Annual International
10. Baker, A.L., Bieman, J.M., Fenton, N., Gustafson, D.A., Melton, A. and Whitty, R.A., "A Philosophy for Software Measurement", The Journal for Systems and Software, Vol. (3), 1990, pp. 277-281
11. Barns D. ,and Mechael, G., "Inheriting Software Metrics", Journal of Object Oriented Programming, 1993, pp.27-34.
12. Basili, V.R., Reiter, R., "Evaluating Automatable Measures of Software Development", Proceedings of Workshop on Quantitative Software Models, 1989, pp. 107-116.
13. Basili, V., and Rombach D.H., "Integrating Measurement into Software Environments", TR-1764; TAME-TR-1, 1987.
14. Bazzana, G., Anderson, O., and Jokela, T., "ISO 9126 and ISO 9000: Friends or foes"? Presented at Software Engineering Standards Symposium, 1993.
15. Belady, L.A., On Software Complexity i: Workshop on Quantitative Software Models for Reliability. IEEE No. TH0067-9, New York, N.Y., pp 90-94, 1979.
16. Beck, Ke., "Extreme Programming Explained: Embrace Change", Boston, MA: Addison-Wesley, ISBN 0-321-27865-8", 2001.
17. Biehl, R.E., "Six Sigma for Software", IEEE Software, Vol. 21(2), pp 68-70, 2001
18. Bieman, J.M., and Schultz, J. "An Empirical Evaluation and Specification Testing Criterion", Software Engineering Journal, Vol. 7(1), 1992, pp 43-51.
19. Bieman, JM., "Deriving Measures of software Reuse in Object Oriented Systems", Technical Report #CS91-112, Colorado State University, Fort Collins/Colorado, USA, 1991.
20. Boddie, J. , "Do We Ever Really Scale Down?", IEEE Software, Vol. 17(5), pp. 79-81, 2000.
21. Boehm, B. W., Brown, J. R., Kaspar, J. R., Lipow, M. L. & MacCleod, G., " Characteristics of Software Quality", New York: Americal Elsevier, 1988.