

## Measuring Process Improvement: Integrating Software Process Assessment With Quality Cost Evaluation

M.P. Neilson and T.P. Rout

Australian Software Quality Research Institute, School of Computing and Information Technology,  
Griffith University, Queensland 4111 AUSTRALIA

### Abstract

This paper reports an extension of a commonly used method for Software Process Assessment by incorporating a detailed breakdown and analysis of quality costs.

The quality cost assessment is based on evaluating a set of baseline activities and corresponding indicators. The results of a case study indicate that the combination of software process assessment and quality cost measurement provides a good indication of the success of process improvement programs.

Keyword Codes: D.2; D.2.2; D.2.8

Keywords: Software Engineering; Tools and Techniques; Metrics

### 1. INTRODUCTION

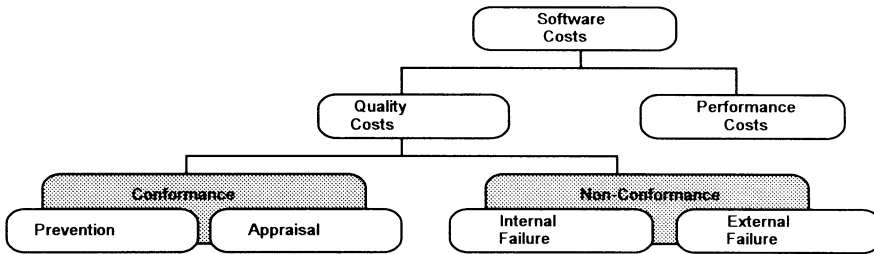
In 1990, Schulmeyer [1] declared that "we are in the midst of a software crisis", following a Price Waterhouse [2] study that revealed non-conformance costs accounted for over 50% of total effort. Schulmeyer [1] called for a "complete restructuring of the software development process" suggesting that the fundamental philosophies of Total Quality Management could be applied to software.

TQM focuses on the process, describing an overall philosophy of continuous process improvement, working to the goal of continually improving the development process. The key notion of TQM is that increasing quality will decrease development costs and consequently increase productivity. Productivity increases as quality costs decrease because of less rework, fewer mistakes and fewer delays. Customers receive higher quality products at a lower cost causing the organisation to dominate the market.

Quality costs are an effective means of controlling the process, ie. quantifying the disadvantages of poor quality and the benefits of quality improvement.

The International Standard for Quality Vocabulary [3], provides the following definition:

*"Quality Cost: the difference between the actual cost to a company of making and selling products and the cost which would exist if there were no failure, and no possibility of failure, of the product to conform to specification during manufacture and use. Quality cost is the sum of prevention, appraisal and failure costs".*



- |   |  |  |   |
|---|--|--|---|
| <ul style="list-style-type: none"> <li>•Quality Planning</li> <li>•Quality Auditing</li> </ul>  | <ul style="list-style-type: none"> <li>•Design Appraisal</li> <li>•Receiving Inspection</li> </ul>   | <ul style="list-style-type: none"> <li>•Scrap</li> <li>•Replacement, rework or repair</li> </ul>   | <ul style="list-style-type: none"> <li>•Complaints</li> <li>•Warranty Claim</li> </ul>                              |
| <ul style="list-style-type: none"> <li>•Assuring vendor/subcontractor quality</li> <li>•Reviewing and verifying designs</li> <li>•Process Control Engineering</li> </ul>  | <ul style="list-style-type: none"> <li>•Inspection and Testing</li> <li>•Inspection and Test Equipment</li> <li>•Materials consumed during inspection and testing</li> <li>•Analysis and reporting of test and inspection results</li> </ul> | <ul style="list-style-type: none"> <li>•Reinspection and Retesting</li> <li>•Defect diagnosis</li> <li>•Disposition Determination</li> </ul> | <ul style="list-style-type: none"> <li>•Products rejected and Returned</li> <li>•Concessions</li> </ul>             |
| <ul style="list-style-type: none"> <li>•Design and development of quality measurement and equipment</li> <li>•Quality Training</li> </ul>   | <ul style="list-style-type: none"> <li>•Field Performance testing</li> <li>•Approvals and Endorsements</li> <li>•Stock Evaluation</li> <li>•Record Storage</li> </ul>  | <ul style="list-style-type: none"> <li>•Downtime</li> <li>•Downgrading</li> </ul>  | <ul style="list-style-type: none"> <li>•Loss of Sales</li> <li>•Recall Costs</li> <li>•Product Liability</li> </ul> |
| <ul style="list-style-type: none"> <li>•Acquisition, analysis and reporting of quality data</li> <li>•Quality Improvement Programs</li> <li>•Product recall and liability insurance</li> <li>•Planning of Product recall</li> </ul> |  |  |   |

Figure 1: Quality Cost Framework

Based upon this definition, a model of quality costs can be constructed that provides a clear delineation of the components of these costs, based on the *Guide to the Determination and Use of Quality Costs* [4]. Initially, a distinction is drawn between the costs of conformance and non-conformance; each of these can then be further broken down, giving a series of individual quality cost components.

Figure 1 shows a conceptual view of Quality Costs. Costs of Conformance can be divided into prevention and appraisal elements. Prevention costs are costs involved in taking measures to ensure that things are done "right first time"; appraisal costs are costs incurred by checking that things have been done right first time. Non-conformance costs comprise internal failure costs - the cost of fixing failures before the product is released to the customer - and external failure costs - the cost of fixing failures after delivery to the customer. Performance costs are the costs that would be incurred during development if the product was 100% defect free or there was no chance of it being otherwise.

## 2. QUALITY COSTS AS A MEASURE OF PROCESS IMPROVEMENT

Process improvement is a long-term incremental activity [5]. The key challenge is to continuously capture and apply knowledge about the limitations of the current processes with the eventual goal of Zero Defects. The limitations can be captured by carrying out a capability evaluation as a result of a process assessment. Several methods for process assessment exist; the best known and most widely used is the Capability Maturity Model (CMM) developed by the Software Engineering Institute [6].

The CMM is a five stage maturity model, based on Crosby's generic structure of quality maturity and the principles of Total Quality Management. An organisation's process is assessed as characterising one of the five "Maturity Levels", Initial, Repeatable, Defined, Managed or Optimising. Each *maturity level* is composed of several *key process areas*. Each key process area comprises a number of *key practices*, organised into five sections called *common features*. The common features specify the key practices that, when collectively addressed, accomplish the goals of the key process area. The common features are attributes that indicate whether the implementation and institutionalisation of a key process area is effective, repeatable, and lasting; the five common features are: Commitment to Perform; Ability to Perform; Activities Performed; Measurement and Analysis; and Verifying Implementation.

Process improvement can be facilitated by Software process assessment. Process assessment is the measurement and evaluation of processes to identify strengths and weaknesses, enabling continuous improvement. The conduct of an assessment involves a disciplined examination of the processes against a set of criteria to perform within quality cost and schedule. The aim of an assessment is to identify the strengths and weaknesses of current practice and the ability of the process to control or avoid significant causes of poor quality, cost and schedule performance [7]. However, there is currently no way of repeatably or comparably quantifying improvement of the process.

Process assessment models can indicate the route to significant progress, by highlighting strengths and weaknesses, but as Lai [8] noted there are two key elements missing. First, there is a lack of valid process-related metrics, especially those that can be used with the CMM; and secondly, there needs to be an economic and business case model of process improvement again based on the CMM. Improvements can be achieved in two ways; by increasing gains and/or reducing costs although "the biggest cost reductions can be gotten by determining what we are spending now for the costs associated with defects, and constantly measuring and monitoring to determine new cost of quality opportunities to reduce [9]". A quality cost assessment framework that would enable valid comparisons across organisations to be made would satisfy both these needs.

The first step toward quantitative analysis is identifying the good and bad practices - process improvement "requires both planting good practices and weeding out bad practices" [10]. This can be achieved by a process assessment. The second step is to put into place the necessary mechanisms to gain statistical control of the process [10], [11]. This can be achieved by measuring Quality Costs.

Software development organisations are already using process assessment methods and quality cost data to measure process improvement and highlight areas for improvement [12]. In 1993 Raytheon reported that they had evolved from a level 1 (on the CMM 5-point scale of capability) in 1988 to a level 3 in 1993. They reported a continuing decrease in the cost of rework. Boehm [13] estimated that eliminating rework might reduce software costs by 30 to 50 percent. During the last four years Raytheon have reduced their rework costs by 30 percent from 41 to 11 percent. However, these measures of improvement are not comparable across organisations and therefore there is no way of valid benchmarking to be carried out.

We have developed a framework that provides a view of quality costs that is directly linked to an overall assessment of organisational and process capability, therefore providing a repeatable and comparable assessment of quality costs. The development process can be examined and improved,

targeting failure costs and investing in prevention. It is a tool that can be used to gain senior management commitment to quality improvement, - "the purpose of comparisons is to get those moving who aren't moving" [14].

### 3. DEFINITION OF THE FRAMEWORK

The framework supports the PAF approach outlined earlier. This approach highlights the strengths and weaknesses and consequently areas for improvement. The purpose of a quality cost assessment is to focus on areas of high expenditure and to identify potential cost-reducing opportunities and more importantly measurement is the first step to control and improvement.

Baseline activities were identified for the quality cost elements of the conformance category. Initially, the goals of the key process areas of the CMM and the activities that included the intent of ISO 9000 from the Trillium model [15] were mapped onto the quality cost elements. The goals of the CMM are a summary of the key practices and are used to determine if an organisation or project have effectively implemented a key process area. The goals were chosen as opposed to the key activities to maintain the simplicity of the framework. Like the Trillium and the Bootstrap models, the intent of ISO 9000 was included because it contains guidelines for an organisation wide quality management system, providing a solid foundation for improvement.

A pilot study revealed that the level of granularity in the framework was inadequate and did not cover specific issues. It was necessary, therefore, to consider a lower level of detail within the CMM in order to achieve the desired level of granularity. Not all of the Common Features contain practices that can be mapped clearly and unambiguously to cost indicators; after investigation, it was decided to restrict the framework to the common feature, "Activities Performed". The ISO 9000 activities from Trillium were found to be satisfactory and were retained.

A set of indicators were defined for each activity, based on the [16] measurable process attributes: time, effort and number of incidents. These indicators are used to determine the quality cost of each activity. Each indicator is classed as:

- a full quality cost,
- a partial quality cost,
- subsumed under other costs, or
- a performance cost.

The costs that are partial quality costs are labelled along with the indicator. The cost of some activities are unavoidably subsumed under other indicators. This does not allow the cost of such an activity to be assessed separately. In some instances it was possible to indicate the activity under which the cost is subsumed. Others are not specifically subsumed under one particular activity and consequently can not be specified. The four examples below serve to demonstrate the nature of the framework:

	Maturity Level	Key Process Area	Activity Performed	Indicator	Cost Category	Cost Status
1	Two	Requirements Management	"The software engineering group reviews the allocated requirements before they are incorporated into the software project."	Effort expended by SE group on reviewing allocated requirements	Prevention - Quality Planning	FULL QUALITY COST
2	Three	Intergroup Co-ordination	"The software engineering group and the other engineering groups participate with the customer and end users, as appropriate, to establish the system requirements."	Effort expended by those other than the SE group with customer and end users establishing requirements.	Prevention - Quality Planning	PARTIAL QUALITY COST
3	Two	Software Quality Assurance	"The SQA group's activities are performed in accordance with the SQA plan."	Costs subsumed under indicators for related activity, "A SQA plan is prepared for the software project according to a documented procedure."	Prevention - Quality Planning	FULL QUALITY COST
4	Three	Software Product Engineering	"The software design is developed, maintained, documented, and verified, according to the project's defined software process, to accommodate the software requirements and to form the framework for coding."	None	Performance Cost	NON-QUALITY COST

Effort is of primary importance in measuring process improvement because it is usually the largest element of cost. In the early 1980's the notion of measuring effort to measure project cost was conceived [17], [18]. Recently, the Software Engineering Institute [19] defined the first repeatable framework for measuring effort and cost specifically for the purpose of tracking project costs. The SEI model allows the project manager to track the actual and planned expenditure of personnel resources. The manager then uses other indicators, such as, progress, cost and quality, to interpret the data. This approach was adopted by Hughes Aircraft to assess process improvement [20]. This method is used to control projects and therefore can not be used to control organisation-wide quality improvement. The SEI framework for measuring effort is not applicable for the purpose of measuring quality costs because it does not support the organisation-wide assessment of a baseline set of activities, and because of this would not facilitate repeatability or comparability.

#### 4. AN EMPIRICAL EXAMPLE

The power and usefulness of the combination of an assessment model with measurement of quality costs was investigated through a case study; the results indicate that the added precision of cost measurement gives added weight to the outcomes of assessment.

##### 4.1 Study Description

The case study was conducted in the IT development unit of a large organisation, employing 120 full time staff. This case study was opportunistic, as the researcher was invited by the organisation's quality manager to assess the quality costs of the unit. The three day, cross-sectional snap shot placed some limitations on the validity of the findings and also had some distinct advantages. It was not possible to view a series of projects to gain an indication of what extent the process is institutionalised. However, the importance of this is not paramount because the organisation can not be assessed by averaging a number of projects, the quality costs may be no better than the worst project. Although a detailed analysis of organisational characteristics such as culture, power structure, etc was impossible in such a short time frame, most of the responses to the researcher were automatic, highlighting the consensus issues [21] and while the social barriers were erect, the individuals within the organisation were reacting naturally.

During the three day period, fifteen hours of semi-structured interviewing was carried out. In addition, six hours of informal discussions were held with five other staff besides those involved with the interviewing. The quality manager who initiated the study became the primary contact, giving as much background and assistance as possible. The senior manager provided job descriptions to assist with the selection of individuals for interviewing.

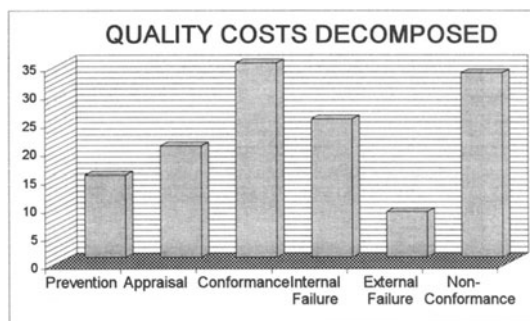


Figure 2: Quality Costs Decomposed

customer trouble reports for the same period that the cost of conformance referenced. These figures did not include machine-time for failure, or any retesting costs. Considering these costs the total cost of non-conformance would be approximately 43% of total development costs. This is still a very conservative figure, it would certainly be much higher taking into consideration the amount of time programmers spend "debugging", if they were to give an honest estimate!

##### 4.2 Research Findings

Non-project specific quality costs were calculated for an 7.5 month period. Project specific quality costs were calculated based on a notional 50 000 person-hour project over 12 months. The assist with the calculations, the organisation provided other relevant figures. Each activity studied was graded as Fully Satisfied (FS), Not Satisfied (NS) or Partially satisfied (PS).

The measurement of the cost of non-conformance was outside the realm of the framework. Figures were given by the organisation based on design and

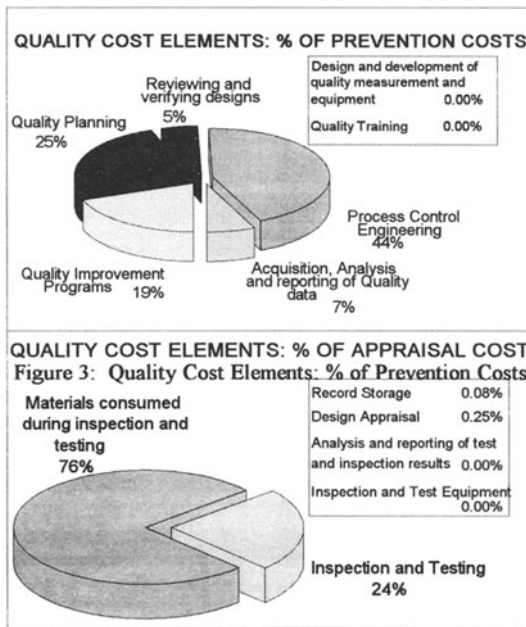


Figure 4: Quality Cost Elements: % of Appraisal Costs

prevention costs are 4.85%.

Figure 4 shows a detailed breakdown of Appraisal costs. The problem with the Inspection and Testing process may be that no effort is expended on Analysis and Reporting of Test and Inspection Results (0%). The same errors could be happening repeatedly and there is no procedure to correct the problems, consequently a continuously high Internal Failure rate.

## 5. DISCUSSION AND CONCLUSIONS

Mandeville [9] defines quality improvement as "the deliberate, planned approach to preventing defects from occurring". There are two approaches to process improvement; analysis and benchmarking [3], [5]. Analysis requires quantitative evidence to determine where improvement needs to be made and where improvements have been made, demonstrating the value of the investment. Benchmarking involves the identification of an "excellent" organisation and striving to match or surpass it.

The main benefit of a framework of this nature is that it provides a view of quality in the context of an overall process assessment which is useful to organisations who are beginning or who have begun a continuous process improvement program. It is important to measure quality costs using the PAF approach, especially for organisations who are currently not competitive. The categorisation will demonstrate dramatically how much is expended on Internal and External failure costs and contrast how little is expended on prevention. Appraisal costs analyse the relationship between prevention and failure costs [9]. The development process can be examined and improved, targeting failure costs and investment in prevention.

The case study shows that the framework is a useful tool for assessing quality costs in software development in an organisational and process capability context. The standardised set of baseline activities ensures that the results are repeatable and comparable.

The other distinct advantage of the framework is that because it is based on a standard set of baseline activities it can be easily migrated from the CMM to any other suitable set of baseline

Figure 2 demonstrates the relationship between each category, Prevention, Appraisal, Internal Failure and External Failure, and as a sum of Prevention and Appraisal - conformance and a sum of Internal Failure and External Failure - non-conformance. The results demonstrate that there are problems with the organisation's processes. The organisation invests large amounts in Appraisal (26.33%) and Internal Failure costs (8.16%) are still high.

Figure 3 shows a detailed breakdown of Prevention costs. Comparisons can be made with the Raytheon study [12] during the level one period where only few prevention activities are carried out. The Raytheon study reported [12] that at level one prevention costs were approximately 5%. This organisation was assessed at a level one and

practices to enable the appropriate comparisons to be made. It is envisaged that the framework could be used over time in parallel with software process assessment as an indicator of process improvement, similar to Raytheon Equipment Division [12]. The advantage of this framework is not only that it can be used for internal quantitative analysis but it can also be a very useful benchmarking tool. It provides a standardised approach to assessing process quality from an economical perspective, that allows comparisons to be made across organisations of the same capability.

## REFERENCES

- 1 Schulmeyer, G., "Zero Defect Software", NY: McGraw-Hill, 1990.
- 2 Price-Waterhouse, "Software Quality Standards: The Costs and Benefits", A Report for the UK Department of Trade and Industry, April 1988.
- 3 ISO 8402 - 1993, "Quality - Vocabulary", Geneva: International Organisation for Standardisation, 1993.
- 4 AS 2561, "Guide to the determination and use of quality costs", Australian Standards Association, 1982.
- 5 Humphrey, W.S., "Introduction to Software Process Improvement", Technical Report, CMU/SEI-92-TR-7, 1992.
- 6 Paulk, M.C., Curtis B. and Chrissis M.P., "Capability Maturity Model for Software, Version 1.1", CMU/SEI-93-TR-24, February, 1993.
- 7 ISO/IEC JTC1/SC7, "The Need and Requirements for a Software Process Assessment Standard", Study Report N944R, Issue 2.0, 11 June, 1992.
- 8 Lai, R., "The Move to Mature Processes", *IEEE Software*, July 1993, pp 14-17.
- 9 Mandeville, W.A., "Software Costs of Quality", *IEEE Journal on Selected Areas in Communications*, February 1990.
- 10 Card, D.N. (ed.), "Understanding Process Improvement", *IEEE Software*, July 1991.
- 11 Jones and Tabberer (1993), "Process Quality Costing adapted to Software Development", *Software Quality J.*, Vol 2, pp 199-208.
- 12 Dion, R., "Process Improvement and the Corporate Balance Sheet", *IEEE Software*, July 1993.
- 13 Boehm, B. W., "Industrial Software Metrics Top 10 List." *IEEE Software*, September 1987: 84-85.
- 14 Crosby, P.B., "Quality is Free", NY: McGraw-Hill, 1979.
- 15 Bell Canada, "Trillium: Telecom Software Product Development Capability Assessment Model", Draft 2.2. Montreal: Bell Canada, July 1992.
- 16 Fenton, N.E., "Software Metrics: A rigorous Approach" London, UK: Chapman and Hall, 1991.
- 17 Boehm, B.W., "Software Engineering Economics", Englewood Cliffs, NJ: Prentice-Hall, 1981.
- 18 DeMarco, T., "Controlling Software Projects: management, measurement, estimation.", Englewood Cliffs, NJ: Prentice-Hall, 1982.
- 19 Goethert, W.B., E.K. Bailey and M.B. Busby, "Software Effort and Schedule Measurement: A Framework for counting Staff Hours and reporting Schedule Information", Technical Report CMU/SEI-92-TR-21, Software Engineering Institute, Pittsburgh, 1992.
- 20 Humphrey, W.S., Snyder B.R. and Willis R.R., "Software process improvement at Hughes Aircraft", *IEEE Software*, July 1991.
- 21 Pressman, R., "Software Engineering: A Practitioner's Approach", 3rd Ed., NY: McGraw-Hill, 1988.