Cost of Quality has remained theoretical in many treatments despite its powerful relevance in software quality. A survey-based approach makes this model an important quality driver today.

Cost of Quality (CoQ) as a software quality model has remained an attractive yet difficult proposition. It is attractive, because it promises to provide decidedly insightful measurements, thus making cost reduction a driver for improving quality. It is difficult, because it demands granular measurement and analysis, making assessment tedious. The good news is that indirect methods employed to leverage the model have helped increase the adoption of Cost of Software Quality (CoSQ). Correlation with CoSQ improvement and CMM® maturity has also been established convincingly in some studies. This document explores the facets of CoSQ and its growing importance in software quality. It suggests a proven approach in making it practically implementable in large software settings.
About the Author

Srinivasa Phanindra Vedula
Srinivasa Phanindra Vedula is a senior Process and Quality Consultant at Tata Consultancy Services. He has extensive experience in the fields of software quality, CMM, CMMI, TPI, and Six sigma. He is a certified Six Sigma Black Belt, and has mentored and coached execution of Black belt projects. He has led and executed consulting engagements for clients in Financial Services, Banking and insurance domain using cost of quality baseline framework outlined in this paper. He is a CSQA and a postgraduate from Indian Institute of Technology, Chennai.
### Table of Content

1. Introduction – Quality for Competitiveness 3
2. Cost of Quality – the Cost that Reduces Cost 3
3. CoSQ Correlates with Software Capability Maturity Levels 5
4. Kick-starting the Journey – Baselining CoSQ 6
5. Case Study: Cost of Software Quality Study for a Financial Services Major in the US 7
6. Concluding Remarks 10
7. References 10
**Introduction – Quality for Competitiveness**

Software quality has come a long way. Having said that, I am probably provoking a section of software makers and recipients who are still skeptical about whether true software quality exists. With quality frameworks and methodologies like CMMI® and Six Sigma being used to the hilt, we somehow think we need to add another perspective to quality, one that can provide a sole indicator for benchmarking and fact-based improvement. Software quality has become more tangible with time and it is interesting to see new dimensions being added to it. In every case though, the quest has been to make it relatable to business outcomes.

Cost of Quality is becoming increasingly important as a driver in business quality. It is also finding serious application in software. This is more so because it provides a unified indicator as well as a driver of business health. Despite the challenge of quantifying the measures involved, it is possible to baseline software quality and benchmark improvements using survey based techniques. This paper demystifies Cost of Software Quality by suggesting a practical approach to baselining CoSQ, which has been successfully used with our global clients. Despite quality generally being an applied discipline, COQ has remained theoretical in many treatments. Therefore, we call this approach “applied cost of quality”.

**Cost of Quality – the Cost that Reduces Cost**

Cost of quality started with the need to reduce cost of meeting client requirements, thereby increasing competitiveness. Over time, it progressed as a philosophy for successful business. Phil Crosby laid the foundation of the current form of COQ in his book Quality is Free (McGraw Hill 1979). He suggested that cost of quality was the sum of two primary costs – cost of conformance and cost of non-conformance. Cost of conformance in turn sums up the cost of appraisal and cost of prevention. The non-conformance cost, on the other hand, has two components – cost of internal failure, that is, the cost of defects that occur before release to market (for instance in development and production), and cost of external failure, that is, the cost of defect that occurs after release to market.

In the case of software, these components can be illustrated as hereunder:

**Cost of Appraisal:**
These are costs incurred to determine the degree of conformance to the requirements. The typical activities of appraisal are
- Review of code and other artifacts
- Purchased software testing
- First iteration of integration and system testing
- User Acceptance testing.
Cost of Prevention:
These are costs incurred to prevent poor quality by keeping failure and appraisal costs to a minimum. The typical prevention occurs through
- Casual Analysis and defect prevention
- Quality planning and software quality assurance
- Quality training on tools and techniques
- Process improvements
- Code and component reuse
- Training on design review

Cost of Internal Failure:
These are costs associated with defects found prior to delivery of software to the customer. The typical activities under this head are
- Pre-release defect management
- Rework
- Re-reviews
- Retesting.

Cost of External Failure:
These are costs associated with defects found after the software is released to customers. Typical activities are:
- Cost of pre-enhancement maintenance and support after release
- Reworks and defect management of field errors
- Warranty redressal

Cost of Execution
These are costs incurred to build the software right first time or the cost of construction, also expressed as percentage of project cost.

Therefore,
Cost of Execution (%) = 100 – Cost of Quality (%)

Cost of quality provides a means to quantify quality cost in business terms. It therefore helps in identifying opportunities for cost reduction. The underlying philosophy is that by cutting down on the cost of quality, one is able to stay more competitive in the market and reduce opportunity cost of losing out to competition. Therefore, in the highest performance state, the effective cost of quality to business is minimal. It emerges as a powerful driver and, at the same time, carries the simplicity of providing one unified metric to judge the health of the line of business. By providing distribution of quality cost across the four components of it, it helps in understanding the direction of improvement focus. For instance, if the cost of appraisal is high compared to that of prevention, one would focus on finding improvement opportunities in defect prevention (such as in design, coding best practices).

The underlying philosophy is that by cutting down on your cost of quality, you are able to stay more competitive in the market and reduce opportunity cost of losing out to competition. Therefore, in the highest performance state, your effective cost of quality to business is minimal. It emerges as a powerful driver and, at the same time, carries the simplicity of providing one unified metric to judge the health of the line of business.
CoSQ Correlates with Software Capability Maturity Levels

The Cost of Software Quality (CoSQ) has a good correlation with capability maturity of the organization, and can be explained by Knox’s model for CoSQ (Modeling the Cost of Software Quality, Stephen T. Knox). Knox used the CoQ model developed in the manufacturing environments and extended it across the SEI CMM® to produce a theoretical CoSQ model. CMM® today is a well-accepted framework to assess maturity of software delivery capability. Thanks to its wide adoption, we have extensive data available on overall industry maturity in terms of CMM® levels. Knox predicted that CMM® Level 1 units have 60% CoSQ and he further hypothesized that CoSQ reduces by 67% at level 5. Knox was able to map the four components of CoQ with the CMM® maturity levels. His finding suggested that the ratio between conformance and non-conformance cost at level 3 would typically be 0.5. In the study, the overall CoSQ exhibits a downward trend and an improvement of cost of execution. Thus, the high maturity organizations get a return of 85% on marginal cost of quality (return on penny spent).
An instance of this is seen in a study by Raytheon Electronic Systems (RES) published in Cost of Software Quality, Herb Krasner, SASPIN, September 1999. It successfully proved that the theoretical CoSQ model is a very good estimator of cost elements and correlates well with CMM® maturity.

Cost of Quality, proportionate to cost of Execution, reduces with higher levels of maturity

**Kick-starting the Journey – Baselining CoSQ**

Establishing CoSQ in an organization is a challenging task. This is because the cost elements that are required for CoSQ are highly fine-grained and demand extensive data. It is not practical to track and classify the project cost data at a granular level. If at all an organization attempts to do so, it would run a high risk of being entrapped in a regimented and ritualistic environment impeding efficiency and ingenuity. Therefore, we suggest a survey-based approach for base-lining CoSQ. Our experience with this approach shows that it can serve as an effective indicator of the direction in which improvement efforts should be focused. The following case study describes a methodology for implementing this approach.
Case Study: Cost of Software Quality Study for a Financial Services Major in the US

TCS IT Process consultants helped in establishing cost of quality baseline using a survey based approach. Under a process reengineering initiative using CMMI® framework, a cost of quality model was designed to estimate the potential savings. The survey inferences were further validated with the qualitative findings during gap analysis.
The following is a part of the questionnaire that was deployed to evaluate the cost of appraisal:

Q. If you are managing a single project, indicate data across the given categories. If you are managing multiple projects, indicate the percentage distribution across the categories.

<table>
<thead>
<tr>
<th>Category</th>
<th>Measure</th>
<th>Design document walk-through</th>
<th>Code walk-through</th>
<th>Specifications walk-through</th>
<th>Test planning, strategy, test case and test bed preparation</th>
<th>Training to testers on tools and defect management</th>
<th>Functional testing</th>
<th>Regression testing</th>
<th>User acceptance testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small projects</td>
<td>% of development effort</td>
<td>% of development effort</td>
<td>% of development effort</td>
<td>% of development effort</td>
<td>% of development effort (First time effort only)</td>
<td>% of development effort</td>
<td>% of development effort</td>
<td>% of development effort (First time effort only)</td>
</tr>
<tr>
<td></td>
<td>Medium projects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Large projects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Very large projects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Result:**

![Average Cost of Quality Distribution](image)

The results showed disparate performance across project types. One of the project types (type 2) showed high cost of prevention. The analysis suggested that this project type had high cost of appraisal to cost of prevention ratio (2.6:1). The interpretation was that opportunities for improvement exist in design and development.
Phase analysis of CoSQ components indicated high injection of defects in the analysis and design phase of large projects, leading to higher cost of prevention during building phase. The average spent on defect prevention during the building phase was significantly higher than industry average.

On further analysis of the defects in building phase, almost two-thirds of the defects were found to be requirement related. This suggested potential opportunities for improvement in analysis and design.

Word of Caution

Since the methodology is based on a survey questionnaire, one needs to guard against measurement errors associated with survey questionnaires. These errors can typically be controlled by paying careful attention to:
- Choice of appropriate representative sample
- Customization of questionnaire for the organization
- Facilitation sessions before administering the survey
Concluding Remarks

Cost of Quality has emerged as a tool for looking at quality from the perspective of enhancing competitiveness. By cutting down the cost of quality, we are improving the overall health of the line of business, which in this case is software. Its strength lies in its ability to provide a unified benchmarking metric that can be further analyzed to identify focus areas for improvements, which otherwise would remain obscure. For instance, a high cost of appraisal may reveal a reactive approach to quality where sufficient attention is not being paid to preventing defects to occur in the first place. However, challenges lie in the implementation of theoretical models and mapping those with the granular activities of the software development lifecycle to gather real data. The survey-based approach to establish the CoSQ financial baseline in IT organizations is a viable and proven alternative in this respect.

References

3. Software Process Improvement the Road to a Professional Software Company SASPIN Nov 2002
4. Cost of Software Quality Herb Krasner SASPIN Sep 1999

Note: CMMI® and CMM® are registered trademarks of Software Engineering Institute, Carnegie Mellon University
About TCS IT Process and Service Management Consulting

IT Process and Service Management Consulting partners with organizations to enable IT process improvement aligned to business needs. Leveraging TCS' experience in successfully adopting multiple frameworks like CMMI®, ISO-9001, ITIL®, P-CMM®, BS15000, BS7799 and Six Sigma, the consulting group provides services with a practitioner's approach to assessing, defining, improving and deploying IT processes.

Service areas include:
- CMMI® Appraisal Services and Training
- ITIL deployment and assessment
- Agile development for Distributed Environment
- Six Sigma based improvements
- Quality Improvement Initiatives
- Digitized enablers, which reduce 'Time to Deploy' and sustain productivity for process

About Tata Consultancy Services (TCS)

Tata Consultancy Services Limited is an IT services, business solutions and outsourcing organization that delivers real results to global businesses, ensuring a level of certainty no other firm can match. TCS offers a consulting-led, integrated portfolio of IT and IT-enabled services delivered through its unique Global Network Delivery Model™, recognized as the benchmark of excellence in software development.

A part of the Tata Group, India’s largest industrial conglomerate, TCS has over 94,000 of the world’s best trained IT consultants in 47 countries. The company generated consolidated revenues of US $4.3 billion for fiscal year ended 31 March 2007 and is listed on the National Stock Exchange and Bombay Stock Exchange in India. For more information, visit us at www.tcs.com

Contact us at
global.consulting@tcs.com

www.tcs.com