Measuring the Impact of Quality Improvement in a Software Company

Cătălin AFRĂSINEI
Babeş-Bolyai University of Cluj-Napoca, Romania
E-mail: cafrasinei@yahoo.com, catalin.afrasinei@econ.ubbcluj.ro
Telephone: +40.264 41 86 52/3/4/5, Fax: +40.264 41 25 70

Abstract:

The quality issue is not only a matter of developing and implementing a quality system. It mandatory this system to function precisely on a long term basis. The evaluation of quality impact as a consequence of its improvement is a scary thing the quality specialists prefer to be apart due to its complexity. That's the reason why the article emphasize on: the need and justification of quality impact evaluation, particularities of quality in software domain generated by its specificity, what evaluation of economic effects means in the context of a quality improvement particularly in a software company, a proposed method to calculate the impact of quality (on the costs structure), a practical example of how the method should be used and the results interpreted based on two simulated case.

Keywords: *quality improvement, quality management, software industry, quality impact evaluation, measure quality.*

JEL Classification: L23, M10, M15

Introduction: The need of quality impact evaluation

Whatever is the economic domain we talk about, the need to provide a highly quality output is more than a reasonable objective. It's almost out of discussion the necessity to guarantee, certify the quality of products or services they offer to customers and this is possible by creating, developing and continuously improving a quality management system which covers and involves the company as a whole.

The literature generously offers a lot of information of what quality and quality system is, the most important quality principles, procedures and guideline of implementing a quality system, advantage and even disadvantages it could bring out. Everything is important and the specialists benefit from a wide range of books and articles which discuss all these topics and more others related to quality

One of the aims of this article is to make the reader to realize that the road doesn't end right after the purpose of implementing a quality system or improving the existing one is accomplished. Because the quality improvement involves efforts and costs, it is realistic to ask yourself about the efficiency of this specific activity, as in the case of the remaining ones.

The expectations of what a quality improvement process brings out could differ from case to case, but it’s sure that they are positive and optimistic in every case. The effort of changing the quality level isn’t negligible and so the effects should be. There are a lot of examples when companies forget to pay a serious attention on the effects of quality improvement actions and even to correlate them with the necessary efforts paid to have this activity completed. They are satisfied when the process is completed and they consider it a success even in this phase. We consider this situation a not completed activity till the moment the impact of that quality improvement is objectively evaluated or measured (whenever is possible). Evaluation doesn’t necessary involves some strict calculations of some indicators, or whatever mathematical formulas to be applied, while measuring does, in a way or other. To measure means to use figures, to calculate, to apply formulas. We admit the fact that measuring the quality impact is a tough task but necessary anytime there are conditions to do it. Based on this idea the article emphasizes on seeking out a simple and practical method of measuring the quality impact and it describes and explain the steps involved and how the results should be interpreted. Nevertheless, the method proposed is way of providing the absolute answer of the problem of measuring the efficiency of quality improvement processes. It is rather a step-forward this purpose to be accomplished and it could be included as a part in a more complex evaluation process.

The example is particularized on a software company but without excluding it from a potential extension to other domains.

We all know the software industry is still one of the most dynamic economic domains and it deserve this position considering its fantastic role on speeding up processes and activities whatever they are and wherever and whenever they appear.

Due to this crucial importance of software domain, which provides the rest of activity domains with specific software products that manage, interpret process and reports data, the need of proving a high quality output is now, more than ever, mandatory since its involvement into economy is more and more deep. As we already stated, this purpose is totally accomplished only when the company has measured the quality impact too, and the results are accepted from efficiency point of view.

1. Quality in software domain, an overview

In order to understand the quality in software domain, we consider necessary to have a good image of its particularities. That’s the reason why we try to summarize some of them in the following.

The software only together with other elements reaches its utility, and all together forms the information technology domain, well-known as IT domain. Software industry is just a part of IT industry, besides other parts like hardware, communication systems and so on. The output of it is generically named “software” or “software application”. The quality in software domain is demonstrated by the quality of its output, which is the software application.

It's important to know that a particularity of IT industry is represented by the powerful implication of the client that chooses (imposes) the wanted configuration of both hardware and software parts of the IT product. Sometime, everything begins from the software which satisfies the client's requirements and the hardware configuration is selected based on the minimum requirements requested by the software for a proper functioning.

In software domain the following elements have to be considered highly important in the quality context: client, human resources, processes, planning, products and expected or forecasted improvements.

Another particularity is related to documentation work, which is essential for the success of any product development in this domain. Documentation is necessary in any stage of the software product lifecycle.

A particularity is the way of specifying the requirements because the characteristics of software products are much different from those of other products.

The most important characteristics of software products are considered the next ones[1]: correctness, maintainability / modifiability, portability, testability, usability, reliability, efficiency, integrity, reusability and interoperability. All of these characteristics become quality characteristics to be followed up when a software product is created and offered to a client.

There are so many quality definitions valid and applicable in software domain. Furthermore, there is a definition which says that quality is hard to be defined, impossible to be measure, easy to be recognized (Kitchenham, 1989)[2]. Their multitude and variety say a lot about the nature of quality.

Quality faces a lot of constraints, especially those ones which are cost related. It's the reason why quality implies sometimes compromises too, those that are accepted by the client through a lower price or cost. Some of the characteristics are easier to be sacrificed than others. For instance, the graphics of an application could be altered in the favor of the operating speed. Quality characteristics are not independent, they have interactions.

David A. Garvin, professor of Harvard Business School[3] has distinguished five main orientations on quality definition: transcendent, product-based, user-based, manufacturing-based, value-based. Each one of these approaches has supporters and critics and each one of them has solid arguments more or less. Here there are some related ideas.

Product-based orientation on defining quality is promoted by economists involved in IT domain. They pointed out that 80% of a software development costs are

[1] Gillies, A., *Software Quality – Theory and Management*, International Thomson Computer Press, London, 1997

[2] Hoyle, D., *ISO 9000 Quality System Handbook*, Reed Educational and Professional Publishing Ltd, London, 1998

[3] Walmüller, E., *Software Quality Assurance – A Practical approach*, Prentice Hall International, London, UK, 1994

generated by maintenance activities[4].

Partisans of this orientation have been criticized due to the fact they neglected the quality dependency by a reference system, established in accordance to the client's requirements.

Quality defined as the conformity with the user's requirements (user-based orientation) tends to „intimidate" some of the software product designers and their reaction might be toward criticizing the users in terms like: „people outside this place cannot do what is right", „they don't understand the quality of software products", „they, surely, don't know what they want"[5].

We can conclude that quality is multidimensional and, regardless the orientation, we consider that it only has a single final validation: the one given by the existing client or the level of how much the product is attractive for the potential client.

2. Evaluation of the economic effects of implementing the quality management system in software domain

Due to implementation of the quality management system according to ISO 9001:2000 standard, a series of benefits are quite rapidly achieved by the most organizations. These initial benefits are generally because of a better organizing and communication within the organization.

When a company adopt ISO 9001:2000 standard, it has to be preoccupied by several things like customers' satisfaction and continuous improvement of products, personnel, quality management system and the whole business. Continuous improvement is a process of increasing the effectiveness and efficiency of the organization on carrying out its policy and objectives in quality domain.

Besides the multitude of benefits, we can imagine, generated by quality systems, partly easy to be measured, partly not, we are still looking for „touchable" proofs for quality system efficiency. **Cost analysis,** could be a one of the ways but it is not enough to run a simple comparison between the amount of costs before and after the process of quality system implementation. We propose a more comprehensive approach and not so complicated to be applied in practice. The implementation of a quality system which involves the entire organization with its activities, processes and functions has direct impact on the **structure of costs** and has at least the same importance as the overall decreasing of costs. Modification of costs structure right after the implementation of the quality system could become an indubitable proof of the usefulness of creating such a system. Taking the case of a software company the efficiency and effectiveness of a quality system is proved, above all, by the savings on the costs engaged into the software creation. It's better the expectations to be not so high at the beginning. Having a high level of

[4] Gillies, A., *Software Quality – Theory and Management*, International Thomson Computer Press, London, 1997

[5] Gillies, A., *Software Quality – Theory and Management*, International Thomson Computer Press, London, 1997

consistence on keeping and even improving continuously the activities in the light of quality management principles will surely reward the effort paid.

In order to determine the costs involved in such a process, from organization’s perspective, they can be structured in 2 categories: **controllable** and **uncontrollable costs.** In the next table we exemplify some categories of controllable and uncontrollable costs that could exist in a software company. Part of them might be present in most of the companies but some of them represent the specific of each one and they have to be identified with responsibility and persistence because they individualize the organization.

Normally, an organization is able to identify the situations which determine uncontrollable costs and it applies plans to avoid such abnormal incidents and the generated costs are accepted as a normal activity risk in the organization’s behalf. In these circumstances companies could make an insurance which covers this type of risks.

According to our opinion, most organizations focus their attention only to controllable costs aiming on their identification and on finding out the ways to decrease them. This demeanor is totally explainable and justifiable considering the fact that controllable costs have the chance to be “controlled”, managed whereas the uncontrollable costs are something that is controlled by someone else. Despite of this, **it’s important to remember that uncontrollable costs affects the company, they have influence on its activities, processes and even on controllable costs**. Having this idea in mind, managers should try harder to estimate, to forecast those events which will eventually generate this type of costs in order to prepare better measures, decisions and finally have the best position the company could get.

Examples of controllable and uncontrollable costs in a software company

Table 1

CONTROLLABLE COSTS	UNCONTROLLABLE COSTS
Costs generated by implementation errors	Costs of energy price increasing
Costs of inappropriate contract modification	Costs generated by random external events
Costs of errors on specifications	Costs of complying with specific regulations
Costs of insufficient prototypes testing	Costs of negative fluctuation of the interest rate
Costs of inappropriate multiplication and storing	Costs generated by important modifications on software market
Costs generated by confusing information included in user manuals	Costs generated by some client requirements

Of course, the impact of an existing and well functioning quality system can be straightly seen on the controllable costs but it also creates a better shield for the company to stand up in front of external factors which generates uncontrollable costs.

Going back to the idea of analyzing the structure of costs before and after the quality system implementation or anytime we apply some changes on it, it's necessary to consider the following classification of quality based costs, proposed by Feigenbaum[6] , probably the most well known classification:

- *Prevention costs* – costs generated by the activities carried out to prevent or reduce defects;
- *Evaluation (or identification) costs* – costs generated by the activities of assessing the level of product conformity with the established requirements (it's about defects identification, in other words);
- *Costs of internal defects (errors)* – costs appeared as a result of defects identified before the delivery of software application;
- *Costs of external defects (errors)* – costs appeared as a result of defects identified after the delivery of software application.

According to the same author, first two categories form **costs of conformance**, while the last two determine the **costs of non-conformance.**

Quality costs have to be identified both for each stage of the software product lifecycle and at all organization's levels.

Generally, identification of all four cost categories should be quite clearly done, but in situations when this is not so visible, solution is to identify some criterions which allow an approach compatible with quality management principles, and keeping no contradiction with accounting principles, such as[7]:

- Any form of planning represents a prevention activity;
- Any activity which assures integrity of testing represents a prevention activity;
- Preparation of control mechanism and procedures of defects treatment and elimination is part of prevention activities category;
- Design, implementation, maintaining and improvement of a quality management system represents a prevention activity;
- Re-evaluation occurred after correction actions applied on previously appeared problems represents an activity of defects identification;
- Cost of low-valued equipments such as measuring tools represent evaluation costs of the year they were purchased;
- The cost of production, development, installation equipments (such as PCs in software industry) which are used both for evaluation activities and for defects identification has to be allocated based on time consumption individually determined for each of the two activities;
- Costs of activities designated to maintain under control some defined and accepted standards represent preventive costs.

[6] Feigenbaum, A. V., *Total Quality Control* (3 ed.), New York, New York: McGraw-Hill, 1991, p. 111

[7] Parker, G.W., *Costurile calităţii,* Editura Codecs, Bucureşti, 1998

Any organization which succeeds on implementing a quality management system has the following objectives, from costs perspective:

- ✓ Decreasing of total quality costs
- ✓ Decreasing of costs caused by defects to a level significantly lower than prevention and evaluation costs level
- ✓ Decreasing of costs of external defects to a level considerably lower than the costs of internal defects

For a better understanding of what we've meant to say, let us take a hypothetical case of a software firm which implements a quality management system. Initially, the structure of the four cost categories might be this one:

- Prevention costs: 10%
- Evaluation costs: 30%
- Internal defects costs: 20%
- External defects costs: 40%

The values might be calculated as percents from total costs.

We can observe that the costs of external defects are significant considering their negative impact on the company's image and relation with its clients. A much better layout of these cost categories in a decreasing order of their values should be: prevention costs, evaluation costs, internal defects costs and external defects costs. External defects costs must have the lowest weight on cost structure.

Let's consider a new situation that might occur after a long enough period of time from implementation:

- Prevention costs: 15%
- Evaluation costs: 60%
- Internal defects costs: 10%
- External defects costs: 15%

By analyzing this new situation we could say that the cost structure has been improved due to the fact the percent weights of internal and external defects costs have decreased as a consequence of an important increase on evaluation, control and testing costs. This proves that much attention has been paid to assure a higher level of software product conformity. However, the above hypothetical case implies some negative aspects too:

- ⇨ Prevention costs still keep a low weight inside cost structure
- ⇨ Reduction of external defects costs should continue at least till the point where the weight is below the level of internal defects.

Besides the improvement of cost structure is also essential the total of these costs to decrease in order to really talk about the efficiency of quality management system.

It is well known that defects discard and re-align the product at the specified conditions costs proportionally with the moment the defect is discovered. The later the moment is the higher the costs are. At the moment the defect produces a lot of troubles for the client, the problem cannot be reduced only on product re-conformance or replacement. We might talk about side effects such as bad company imagine in customers' perception or even juridical consequences. These image and credibility consequences could be fatal for the organization.

Intangible components, such as clientele loss, clients' dissatisfaction, and difficulties to entry on new market segments could be evaluated through market surveys.

Decisions regarding resource allocation in order to prevent to supply the clients with defect products (software applications) have to take into consideration the following aspects:

- Prevention activities determine decreasing of defects number which generate decreasing of both evaluation costs and other defects related costs;
- Evaluation activities don't reduce the number of defects. They are just identified and analyzed and consequently the evaluation costs and internal costs are increasing but the external costs of defects are decreasing in the same time.

Concluding, in software domain investments in prevention activities are more efficient than those for evaluation. This is confirmed by the practice which demonstrates that bigger amounts of prevention costs determine important decreasing on total quality costs.

Next figure presents the variation of the cost of an error correction in software domain correlated with the phase of its identification (moment of discovering it).

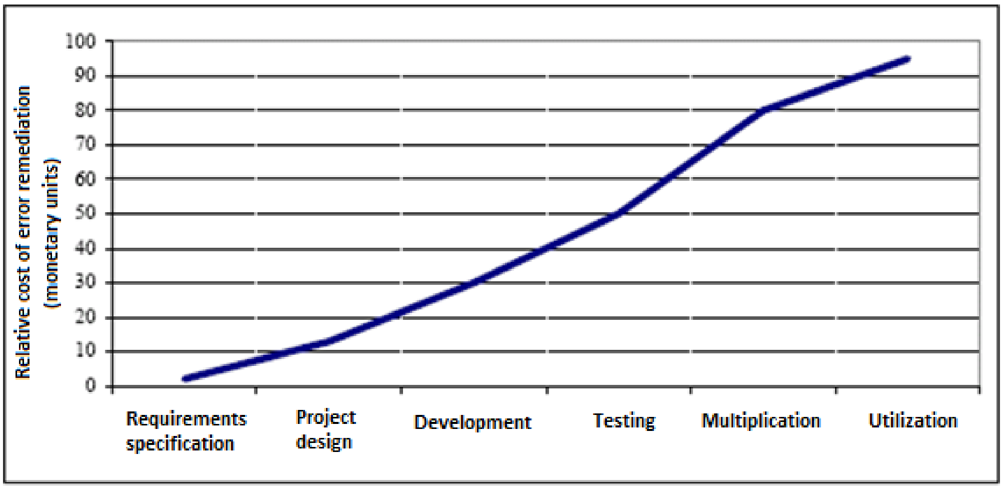

Figure 1 Cost of an error correction in the case of a software product

Source: Parker, G.W., *Costurile calităţii,* Editura Codecs, Bucureşti, 1998

Of course the situation presented in the graph comprises some analysis and comments. Those six processes represent indeed the necessary stages a software product should pass from the idea to its final form ready to be used and capable to meet requirements. They have a logical succession (as seen in the graph, from left to right) and they are strongly related. Each of the processes has inputs and outputs and, moreover, the output of a process becomes the input of the next process and so on. Therefore, each process is influencing the next process in the raw after it has already been influenced by the process which precedes it according to the working flow.

However, an error occurred in "requirements specification" phase for instance, which hasn't been discovered quickly, it has the chances to propagate even till the phase of utilization. This is a very bad situation and the impact is very high because of the following aspects, at least:

- ✓ The client is the one who discovered the error (negative perception, negative image);
- ✓ The software has to be multiplied again, not only for that client, but most probably for a very large number of clients who use the same version on application;
- ✓ Phases of design, development, testing have to be partially taken from some point on or even we could talk about re-design, re-development, re-testing if the error is a fundamental one;
- ✓ Human resources might be relocated from other projects so as to establish the normality as soon as possible.

The situation is not so bad (the impact is slightly lower) if the error discovered in utilization phase occurred in multiplication phase. This means that something was wrong into this phase and no one of the upstream phases is affected by this error.

Concluding, we could say that the impact (on both costs and company's image) of a defect depends on the following factors:

- ➢ The phase of the technological flow where it occurs
- ➢ The phase where it is discovered and how big is the "distance" from the phase it appeared. So, we talk about a distance between occurrence and discovering points.

Considering all of the ideas discussed, it still remains the problem of identifying and quantifying the four categories of quality costs (proposed by Feingenbaum) for each of the processes included into the software application production (as seen in the above graph).

3. A possible method of measuring the quality impact

Supposing that a software company, which has implemented a quality management system, is capable to quantify the four categories of quality costs, which are prevention, evaluation, internal and external defects costs, the next step is to observe their structure before and after implementation or before and after a quality improvement process.

It's obvious that the structure of quality costs is going to change. The question is that the new situation is better than the previous one. Taking the hypothetical case presented above, it's clear that the new situation is better even for the fact that the costs of non-conformance (internal plus external defects costs) decreased from 60% to 25%. Having in mind the ideas of economic efficiency, the next question should be: "But how much the new situation is better? How much is the impact of this new situation? Is the effort involved worth?" These are reasonable questions a good management should ask. Hence, we consider that is necessary to look for a way to answer them.

We propose o possible procedure/method to calculate a score of negative impact a quality management system or quality improvement might have on the overall company's costs and image and we could name it Quality Impact Score. Accepting that each of the four quality cost categories has different levels of impact, it's important to establish somehow these levels. The impact of each cost category could be evaluated considering some factors which have importance either for company or customer, such as:

- The amount of money involved
- The impact on customer's perception

Considering these two factors or some more, according to each company specificity and interests, each cost category could be granted with an impact index, as in the next example:

- Prevention cost index: 1
- Evaluation cost index: 2
- Internal defect cost index: 5
- External defect cost index: 10

Considering the first and the forth category from the above example, we can state that the negative impact of the external defect cost is 10 times bigger than the prevention cost. Similarly, the negative impact of internal defect is 5 times bigger than the prevention cost.

Therefore, the total impact score could be calculated starting from the structure of quality costs and weighted with the corresponding impact index, as shown in the following table:

Calculation of Quality Impact Score

Table 2

Quality costs	**Impact index**	**Structure of costs (%)**		**Impact score**	
		Case A	**Case B**	**Case A**	**Case B**
Prevention cost	1	10	15	0.10	0.15
Evaluation cost	2	30	60	0.60	1.20
Internal defect cost	5	20	10	1.00	0.50
External defect cost	10	40	15	4.00	1.50
TOTAL	-	100	100	**5.70**	**3.35**

Interpretation: the smaller the quality impact score is, the better the situation of company is, from quality perspective. Calculation of this score has been made as follows:

For Case A:

$$5.70 = \frac{10 \times 1 + 30 \times 2 + 20 \times 5 + 40 \times 10}{100}$$

where:

- 10, 30, 20 and 40 represent the structure of quality costs (percents),

- 1, 2, 5 and 10 are the corresponding impact indexes

- 100 is the sum of cost category percents (which is 100 all the time in normal conditions)

For Case B:

$$3.35 = \frac{15 \times 1 + 60 \times 2 + 10 \times 5 + 15 \times 10}{100}$$

The values of Quality Impact Score range between 1 and 10. Value 1 is the ***best possible situation*** from quality perspective when prevention costs represent the only quality costs valued at 100%. Then we have:

$$1.00 = \frac{100 \times 1}{100}$$

The worst situation is when external defect costs represent the only type of quality costs. So, they represent 100% of total quality costs. Then we have:

$$10.00 = \frac{100 \times 10}{100}$$

Resuming, we can say that Case B is better from quality management point of view because the quality impact score is closer to 1 than Case A. Going deeper into details, its observable, from the table, the high value of impact score of external defect costs in Case A. That's the reason why the total score is so unfavorable in the first case despite its better position for the first two cost categories, which actually counts much less in the calculation of total impact score.

Concluding, the final impact score will always be closer to its minimum value (best situation) if the costs of conformance (prevention plus evaluation) are predominant in the structure of quality costs.

Conclusions

The article emphasizes on the idea of measuring the impact of quality improvement in the case of software companies. The logic of this article is based on several ideas developed throughout its body:

- there is necessary to mind the importance of evaluating the impact of a quality improvement process to eventually declare it as being a success
- the software domain is a special one because of its specificity. So is

the quality, which has particularities. A part of them have been presented in the text considering the fact they influence the quality framework, the way the quality concepts are developed, implemented, evaluated in software companies

- the quality evaluation process is a complex one and it has several sides and stages. One of them is the evaluation of economic effects of quality improvement. We provide details about this type of evaluation considering its importance in the overall evaluation process. The main impact of quality improvement is visible on the value of costs encountered before and after quality improvement
- the structure of quality costs is also affected by quality improvement. We explained this topic theoretically and by giving numerical examples
- in the final section of the article we proposed a method to evaluate the quality impact based on an score which could be calculated taking into account the structure of quality costs before and after implementation. It is an effort to simplify the evaluation process and to give it a more practical view especially for those specialists implicated into quality problems in software companies and not only.

References

1. Balog, A., 1997. *Analiza statistică şi evaluarea calităţii software*-*ului*. Bucuresti: Editura Calipso 2000
2. Feigenbaum, A. V., 1991. *Total Quality Control (3 ed.)*. New York: McGraw-Hill
3. Gillies, A., 1997. *Software Quality – Theory and Management*, London: International Thomson Computer Press
4. Horch, W., J., 1996. *Practical Guide to Software Quality Management*. Boston: Artech House Inc
5. Hoyle, D., 1998. *ISO 9000 Quality System Handbook*. London: Reed Educational and Professional Publishing Ltd
6. Parker, G.W., 1998. *Costurile calităţii,* Bucureşti: Editura Codecs
7. Walmüller, E., 1994. *Software Quality Assurance – A Practical approach*. London: Prentice Hall International