Using Economics as Basis for Modelling and Evaluating Software Quality

Stefan Wagner
Institut für Informatik
Technische Universität München
Boltzmannstr. 3, 85748 Garching b. München, Germany
wagnerst@in.tum.de

Abstract

The economics and cost of software quality have been discussed in software engineering for decades now. There is clearly a relationship and a need to manage cost and quality in combination. Moreover, economics should be the basis of any quality analysis. However, this implies several issues that have not been addressed to an extent so that managing the economics of software quality is common practice. This paper discusses these issues, possible solutions, and research directions.

1. Introduction

The economics – and more often only the costs – of quality has been a topic in various disciplines [7, 8, 14] and has also been discussed in software engineering [3, 9, 11, 16–18, 20, 21]. All this work shows that economics is a suitable basis to discuss and analyse software quality. There are two main reasons for that:

1. Most software projects are done by companies that have the aim of economic success. Hence, these projects need to be managed on the basis of economic decisions. This includes also decisions w.r.t. software quality.
2. Economics offers the only universal unit that all influencing factors of software quality can be reduced to. "Quality is a complex and multifaceted concept." [10] These concepts can only be related by assigning a value to them.

Therefore, economics and quality of software are two concepts that need to be discussed in combination. In an engineering context, quality cannot be seen as an intrinsic attribute of a product but as the combined influence of various properties of the product on its economics. As pointed out in [5] "project are not interested in 'maintainability' per se; the crucial parameter for them is the 'maintenance effort' (best measured in some currency)." It starts with the goal definition in requirements engineering where quality goals should have an economic reason. Also the use and content of quality models that are used to analyse and improve software quality should be based on economic factors. Finally, decisions in quality assurance and maintenance boil down to costs. For example, maintainability often is defined as "the ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment." [12] This means that the effort needed – and hence the costs – are decisive for this quality attribute.

However, there are several problems and issues in the combination of quality and economics. In our view, there are four main issues that need to be tackled in order to achieve a more thorough understanding of the relationships of quality and economics and bringing methods and techniques into practice:

- Lack of consideration of economics in quality models
- Lack of empirical knowledge in research
- Lack of relevant data in industry
- Difficulties of using monetary units
- Lack of education related to economics and stochastic models

We will detail these points in the following and discuss possible solutions and research directions.

2. Quality and economics

We discussed the reasons why a value-based view on quality should be the main focus w.r.t. software quality. The economic success is the most important goal of the majority of software systems and monetary units are the only common unit of all influencing factors (cf. [21]). What follows from this insight is that software quality cannot be determined by itself or with a single figure. Software quality is inseparably connected to its influence on the economics, i.e. the cost-benefit relation, of the software.

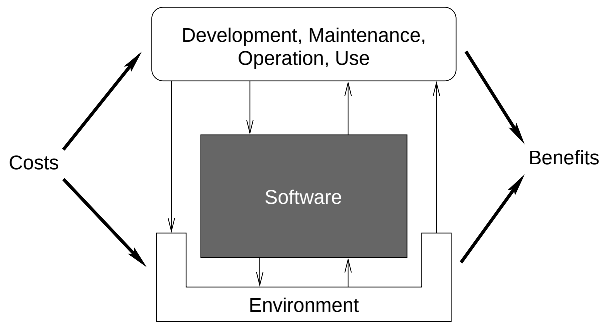

Figure 1. The relationship of quality and economics

Fig. 1 shows this connection graphically. Each software is embedded in some environment consisting of other software components, platform software, and hardware. There are also various activities that are performed on the software:

- The initial *development* of the software
- *Maintenance* of the software consisting of corrective, adaptive, and perfective changes
- The software needs to be administrated during *operation*
- The primary purpose is that the software is in *use* performing its function

These two blocks – the environment and the activities – incur costs and generate benefits in combination with the software. The benefits are often saved costs but can also be independent of earlier costs. For example, the benefits can be shorter production times or new overseas markets because of online marketing. The environment has also an influence on the activities. It depends on whether we consider the software (e.g. an enterprise resource planing system) or the combined hardware/software system (e.g. an airbag control system) as the main product. In the former case, the environment can be seen as an addition to the software that can be cheaper or more expensive depending on the demands of the software. In the latter case we should handle the environment equally as the software and consider the influences on the activities directly.

The quality of the software is then how it influences the activities and its environment in incurring costs and generating benefits. In Fig. 1, the thin arrows represent the quality of the software (and its environment). Only these influences allow to judge whether a software is of "good" quality. This again shows that quality is a multi-faceted concept that lies strongly in the eye of the beholder. It also follows that a single figure or measure will not be able to describe all these influences.

3. Issues

Having discussed the inherent relationship of quality and economics in software systems, we bring up several issues related to this connection. We describe the problem and state possible solutions and research directions.

Consideration of economics in quality models. The available models of software quality aim at decomposing quality along one dimension, e.g. [4, 13]. Activities and properties of the system are intermingled in a single dimension. However, this neglects the relationship of quality and economics identified in Sec. 2. It hampers a clear structure of the model. Moreover, the models tend to suggest that quality is an intrinsic property of the software although – as pointed out above – mainly the influences on the activities performed on the software determine the quality.

In our research group, we have developed a two-dimensional quality model that aims to address this issue [6, 22, 23]. The model contains the facts about the software, its environment and the corresponding process as well as the activities performed on the software. Then these two parts are linked by influences. For example, in the usability model [23], we can model the influence of the number of steps needed to accomplish a task to specific usage activities. This way, we can qualitatively analyse usability and have a basis for quantification.

Empirical knowledge in research. Empirical research in software engineering is well established by now but it has still not gained the importance it deserves. For an economic analysis of software quality, most questions can only be answered empirically. Some questions have been worked on by the empirical software engineering community such as the efficiency of testing and inspection techniques, e.g. [2, 15]. These issues are now better understood. However, the complete relation to economics is still not clear.

Moreover, there are other factors that influence software quality and they need to be identified by field studies and experiments. Another problem is that studies at companies are difficult because the cost data is typically considered as very sensitive, i.e. secret information that must not be given to competitors. We need to find ways to address these issues and to convince companies of the advantages. For example, it allows a better comparability of the own efforts with the domain average.

Data in industry. The main fact that hampers analyses of quality economics in practice is the little data that is available in industry. There are several reasons for that. The main reason is the effort needed for collecting appropriate data. It is often the case that companies might be interested in results about quality and economic analyses but are not willing to invest the necessary effort. Hence, research needs to investigate the economics of quality economics. We need to be able to show to what extent it pays off to collect specific data.

Moreover, in industry, it is not well understood how to use collected data. Again, research needs to clearly show which metrics can be used for what purposes. The goal-question-metric paradigm [1] is clearly a basic step in that direction. Furthermore, we propose that quality models are a way to achieve well-founded reasons for metrics [6, 22]. Another reason for the poor availability of data is that company-internal politics can be involved. In certain situations it might be better for some employees that the economical implications of their decisions are not known. For example, this helps to disguise mistakes. A solution can only be an open communication culture in the companies.

Monetary units. We argued for economics because money is the only possible unit that is universal enough to measure the influences on quality. However, the monetary units are not as universal as we first suggested. The value of money changes over time, with different currencies, and over different locations. The main problem is inflation. How can we build empirical knowledge using a continously changing unit? One answer is surely to use net-values and other methods that allow to inflation-adjust the values. However, this renders meta-analysis even more difficult as it is today (cf. [19]).

On top of this, we have to deal with different currencies that have also ever-changing exchange rates. This also hampers comparability. A solution would be to convert all data into one broadly accepted currency. Moreover, the labour rates can differ even in a single currency. Because human labour costs constitute the main costs in software development, this complicates comparisons strongly. We suggest to agree on a standard date and currency that all data is converted to.

Education. Computer science education is largely governed by its root in formal mathematics. The software engineering reality that also includes economics is not accurately represented there. Very basic techniques such as statistical process control or building stochastic models are scarcely taught. We propose to include mandatory modules of basic economics as well as quantitative and qualitative management in the curricula. This way, graduates will have a better understanding of the relationships between software quality and economics discussed in Sec. 2.

4. Conclusions

Economics and quality are two fundamental concepts in software development that are closely related. A thorough management of one of the two implies also the consideration of the other. There are several problems in research and practice today that hamper the modelling and evaluation of these relations. We described several of these issues, possible solutions, and research directions to address them. We believe that the main issue for research is to build suitable quality models and find ways to found them with a solid empirical basis.

Acknowledgements

We are grateful to our colleagues Florian Deissenboeck and Markus Pizka for many fruitful discussions on the topic.

References

[1] V. R. Basili, G. Caldiera, and H. D. Rombach. Goal Question Metric Paradigm. In J. C. Marciniak, editor, *Encyclopedia of Software Engineering*, volume 1. John Wiley & Sons, 1994.

[2] V. R. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sorumgard, and M. V. Zelkowitz. The empirical investigation of perspective-based reading. *Empirical Software Engineering*, 1(2):133–164, 1996.

[3] B. Boehm, L. Huang, A. Jain, and R. Madachy. The ROI of software dependability: The iDAVE model. *IEEE Software*, 21(3):54–61, 2004.

[4] B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, G. J. Macleod, and M. J. Merrit. *Characteristics of Software Quality*. North-Holland, 1978.

[5] M. Broy, F. Deissenboeck, and M. Pizka. A holistic approach to software quality at work. In *Proc. 3rd World Congress for Software Quality (3WCSQ)*, 2005.

[6] M. Broy, F. Deissenboeck, and M. Pizka. Demystifying maintainability. In *Proc. 4th Workshop on Software Quality (4-WoSQ)*, pages 21–26. ACM Press, 2006.

[7] P. B. Crosby. *Quality Is Free*. Signet, reissue edition, 1980.

[8] A. V. Feigenbaum. *Quality Control: Principles, Practice, and Administration*. McGraw-Hill, 1951.

[9] D. Galin. *Software Quality Assurance*. Pearson, 2004.

[10] D. A. Garvin. What does "product quality" really mean? *MIT Sloan Management Review*, 26(1):25–43, 1984.

[11] W. S. Humphrey. *A Discipline for Software Engineering*. The SEI Series in Software Engineering. Addison-Wesley, 1995.

[12] IEEE Std 610.12-1990. *IEEE Standard Glossary of Software Engineering Terminology*, 1990.

[13] ISO. Software engineering – product quality – part 1: Quality model, 2001.

[14] J. M. Juran and A. B. Godfrey. *Juran's Quality Handbook*. McGraw-Hill Professional, 5th edition, 1998.

[15] N. Juristo, A. M. Moreno, and S. Vegas. Reviewing 25 Years of Testing Technique Experiments. *Empirical Software Engineering*, 9:7–44, 2004.

[16] S. T. Knox. Modeling the costs of software quality. *Digital Technical Journal*, 5(4):9–16, 1993.

[17] D. Raffo, W. Harrison, J. Settle, and N. Eickelmann. Understanding the role of defect potential in assessing the economic value of process improvements. In *Proc. Second Workshop on Economics-Driven Software Engineering Research*, 2000.

[18] S. A. Slaughter, D. E. Harter, and M. S. Krishnan. Evaluating the cost of software quality. *Communications of the ACM*, 41(8):67–73, 1998.

[19] S. Wagner. A literature survey of the quality economics of defect-detection techniques. In *Proc. 5th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE '06)*. ACM Press, 2006.

[20] S. Wagner. A model and sensitivity analysis of the quality economics of defect-detection techniques. In *Proc. International Symposium on Software Testing and Analysis (ISSTA '06)*, pages 73–83. ACM Press, 2006.

[21] S. Wagner. *Cost-Optimisation of Analytical Software Quality Assurance*. PhD Dissertation, Technische Universität München, 2007. To appear.

[22] S. Wagner and F. Deissenboeck. An integrated approach to quality modelling. In *Proc. 5th Workshop on Software Quality (5-WoSQ)*. IEEE Computer Society Press, 2007.

[23] S. Winter, S. Wagner, and F. Deissenboeck. A comprehensive model of usability. In *Proc. Engineering Interactive Systems 2007 (EIS '07)*. Springer, 2007. To appear.