2004-01-01

# Software quality challenges.

Ronan Fitzpatrick
*Technological University Dublin*, Ronan.fitzpatrick@tudublin.ie

Peter Smith
*University of Sunderland*

Brendan O'Shea
*Technological University Dublin*, brendan.oshea@tudublin.ie



## Recommended Citation

Fitzpatrick, Ronan and Smith, Peter and O'Shea, Brendan: Software quality challenges. Proceedings of the Second Workshop on Software Quality at the 26th. International Conference on Software Engineering (ICSE 2004), Edinburgh, Scotland. Published by IEEE.

# Software Quality Challenges


Ronan Fitzpatrick
*School of Computing,*
*Dublin Institute of Technology,*
*Kevin Street, Dublin 8, Ireland.*
*ronan.fitzpatrick@comp.dit.ie*

Peter Smith
*School of Computing and Technology,*
*University of Sunderland,*
*Sunderland SR6 0DD, UK*
*peter.smith@sunderland.ac.uk*

Brendan O'Shea
*School of Computing,*
*Dublin Institute of Technology,*
*Kevin Street, Dublin 8, Ireland.*
*brendan.oshea@comp.dit.ie*



## Abstract

*This paper sets out a number of challenges facing the software quality community. These challenges relate to the broader view of quality and the consequences for software quality definitions. These definitions are related to eight perspectives of software quality in an end-to-end product life cycle. Research and study of software quality has traditionally focused on product quality for management information systems and this paper considers the challenge of defining additional quality factors for alternative domains like the World Wide Web.*



***Keywords**: Research challenges, software quality definitions, quality perspectives, end-to-end product life cycle, strategic drivers, additional quality factors for WWW.*


## 1. Introduction

Research relating to software quality is typically rooted in the study of product quality factors and the usability of those products in a context of use [1], [2] and [3]. During this research and study emphasis on quality assurance and measurement is limited to this product perspective. Furthermore, the domain in which quality is measured is limited to that of Information Systems (IS). Insofar as it relates to the IS domain, the paper first considers definitions of quality and other related issues.

As evidenced by the needs of eCommerce it is also necessary to broaden the study of software quality to embrace other domains like the World Wide Web (WWW). In this domain, product quality necessitates the study of additional quality factors which address access, interaction and navigation. Furthermore the owners of eCommerce solutions have new expectations that they will gain competitive advantage from their sites and this introduces further perspectives of software quality beyond that of product quality.

Combining both of these, this paper presents a number of challenges for the software quality community. The paper is based on many years experience of both teaching and researching software quality and is of interest to both academic and practitioner alike.

Section 2 considers the focus of the definitions of quality and especially software quality. Section 3 revisits external and internal quality and examines understanding of quality-of-use. Section 4 highlights the many different perspectives of quality in an alternative end-to-end software product life cycle model. This section also highlights an over emphases on software testing to the detriment of managing software quality. Section 5 explains the need for redefining quality in the light of evolving technology and in particular eCommerce.

## 2. Quality defined

There are many different definitions of quality [4] to [11]: (Crosby, 1979; Deming, 1986; Feigenbaum, 1961; Ishikawa, 1985, Juran, 1989; Oakland, 1993; Shingo, 1987; Taguchi *et al.*, 1987). It is typically defined in terms of *conformance to specification* and *fitness for purpose*. Figure 1 shows a number of acknowledged definitions.

| Acknowledged authority | Definition or interpretation of quality |
|---|---|
| Crosby, 1984, p60 | Conformance to requirements. |
| Deming, 2000, p168/9 | Quality can be defined only in terms of the agent. Who is the judge of quality? Deming continues, "*The problems inherent in attempts to define quality of a product… were stated by the master Walter A Shewart (1986, Ch 4) viz the difficulty in defining quality is to translate future needs of the user into measurable characteristics, so that the product can be designed and turned out to give satisfaction at a price that the user will pay*". |
| Feigenbaum, 1961, p13 | The composite product characteristics of engineering and manufacture that determine the degree to which the product in use will meet the expectations of the customer. |
| Juran, 1988, p11<br>1989, p15 | Quality is product performance, quality is freedom from defects, quality is fitness for use.<br>Fitness for use |
| Oakland, 1993, p4 | Meeting customer's requirements. |
| Shingo, 1986, p11 | Zero defects. |
| Taguchi, 1987 | Product quality is determined by the economic loss imposed upon society from the time a product is released for shipment. |

**Figure 1 – Definitions of quality.**

There are difficulties with definitions that focus on *conformance to specification* and *fitness for purpose*. In the first instance it follows that if there is a deficiency in the specification then there will be a deficiency in the quality, yet the definition would imply that *conformance to the specification* will produce a quality product. This is not the case and an inferior or deficient specification will not produce a high quality product. *Fitness for purpose* can also be challenged along the same lines. For example, there are many types of motor cars that are fit for the purpose of transportation of two to four individuals from A to B. But they are not all Rolls-Royce quality cars. So, *fitness for purpose* does not fully define quality either.

International Standards Organisations also define quality. A selection of International definitions is shown in Figure 2.

| Standards Body | Definition of quality |
|---|---|
| German Industry Standard DIN 55350 Part 11 | Quality comprises all characteristics and significant features of a product or an activity which relate to the satisfying of given requirements. |
| ANSI Standard (ANSI/ASQC A3/1978) | Quality is the totality of features and characteristics of a product or a service that bears on its ability to satisfy the given needs. |
| BS 4778,1987 (ISO 8402, 1986) | The totality of characteristics of a product or service that bear on its ability to satisfy stated or implied needs. |
| IEEE Standard (IEEE Std 729-1983) | **a** The totality of features and characteristics of a software product that bear on its ability to satisfy given needs: for example, conform to specifications.<br>**b** The degree to which software possesses a desired combination of attributes.<br>**c** The degree to which a customer or user perceives that software meets his or her composite expectations.<br>**d** The composite characteristics of software that determine the degree to which the software in use will meet the expectations of the customer. |
| ISO/IEC 9126 (1991) | The totality of features and characteristics of a software product that bear on its ability to satisfy stated or implied needs. |

**Figure 2 – International standards definitions of quality.**

When the quality relates to software quality it is mainly defined in terms of characteristics of a product and its use. There are two very important points in these definitions. The first is, they emphasise the product and in the case of software this is the application delivered to the purchaser. The second is that they introduce the desirability of measurement by using words like totality and degree. This is in keeping with a natural description of high or low quality which in scientific terms might equate to a scale such as 0 to 100. In the domain of Information Systems, quality is limited to measuring the attributes (the quality factors) of the software product and measuring its use [3]. This is the narrow view of quality which only addresses quality-of-process, quality-of-product and quality-of-use.

A broader view of quality is suggested by the founding father of the Japanese quality movement, Kaoru Ishikawa [7]. His view of quality is shown in Figure 3.

| | |
|---|---|
| Ishikawa, 1985, p44/5 | In a series of definitions relating to quality control he refers to products which can "*satisfy the requirements of consumers*" This he explains should be "*Narrowly interpreted to mean quality of products*". He continues "*that Broadly interpreted quality means quality of work, quality of service, quality of information, quality of process, quality of division, quality of people including workers, engineers, managers and executives, quality of system, quality of company, quality of objects etc. To control quality in its every manifestation is our basic approach*". |

**Figure 3 – Ishikawa's broader view of quality.**

On this basis, it follows that limiting software quality to the process by which the product is built and to its usability is too narrow a view and that there are a number of perspectives of quality (some of which are not widely researched). Eight perspectives are represented on the newly extended Software Quality Star mark II as illustrated in Figure 4.

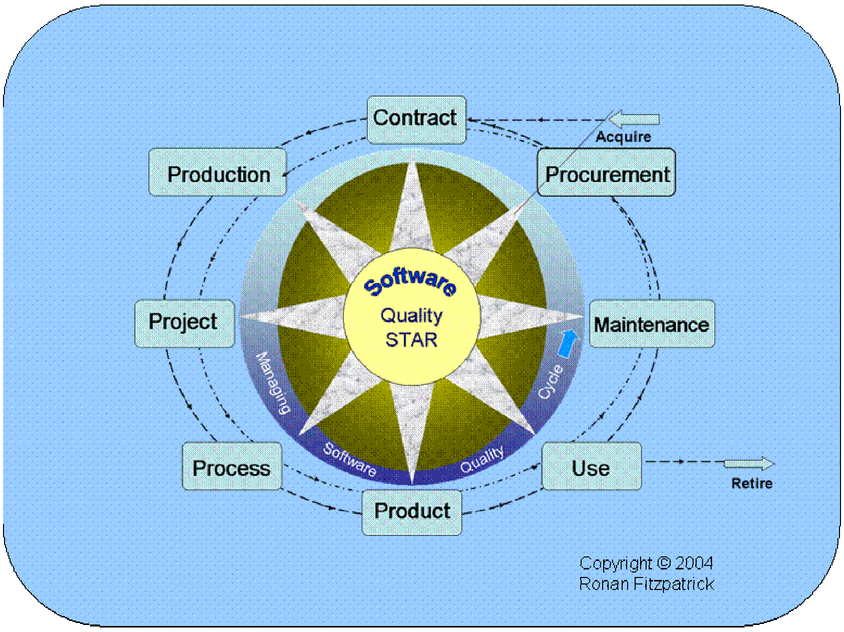


**Figure 4 - Software Quality Star mark II (*SQ-Star*).**

The Software Quality Star mark II is an enhanced version of the original model, [12]. Its original motivation was to illustrate the principal points of focus in ISO 12207 [13] which relates to software life cycle processes. Mark II is enhanced to incorporate end-to-end perspectives together with domains like the World Wide Web which are additional to the Management Information Systems domain.

The eight perspectives in the Software Quality Star are quality-of-procurement, quality-of-contract, quality-of-production, quality-of-project, quality-of-process, quality-of-product, quality-of-use and quality-of-maintenance.

So, it is appropriate to step back and consider quality on a higher level. It can easily be argued from the definitions in Figures 1, 2 & 3 that quality is a measure of something (other than product characteristics) relating to the different perspectives and this paper proposes that at the higher level *quality is a measure of excellence*. The excellence should then be quantified for each perspective. For example, in the case of quality-of-product, the excellence will relate to product external and internal quality factors. In the quality-of-production perspective the excellence will relate to the producer considerations and in the perspective of the owner the excellence relates to procurement and issues like value for money and competitive advantage. So, software quality could be defined in terms of a measure of excellence in the perspectives of the end-to-end software product life cycle.

## 3. External and internal quality

Software quality is typically divided into external and internal quality. It is appropriate to revisit these divisions to see if the quality factors in each category are correct. The motivation for this is that external quality is the category that directly impacts the user and this paper argues that everything that impacts the user is more accurately named usability. Therefore, all of the external quality factors (all of which impact the user) should be collectively referred to as usability. This would necessitate renaming the original usability factor, and ease-of-use seems to more accurately describe it. Interpreting the term usability as meaning anything that impacts the end user is a more natural interpretation of the term than has been used heretofore.

A similar challenge exists in relation to internal software quality and it might be more meaningful to collectively name all of the internal factors using one name. Typically they might be styled maintainability or evolvability: one word that encapsulates corrective, adaptive, perfective and progressive maintenance.

## 4. Quality in life cycle models

The third challenge addressed in this paper focuses on quality in the life cycle.

Popular conceptual system-life-cycle models are software engineering focused with processes mainly centred on the creation of the software product. That is, they address quality-of-process, quality-of-product and quality-of-use in a context of use. But the broader view of quality dictates that a life cycle that focuses only on *software development* is insufficient and that a full end-to-end *software product* life cycle is required. Such a model would embrace quality from product conception through to product retirement and would address all of the quality perspectives of the Software Quality Star – Figure 4. That is, quality-of-procurement, quality-of-contract, quality-of-production, quality-of-project, quality-of-process, quality-of-product, quality-of-use and quality-of-maintenance.

This end-to-end model also addresses the fact that the word quality is not mentioned in the popular conceptual system-life-cycle models. Expressions like validation and verification or test or evaluation are used but quality as a focus of management during the life cycle is not given the significance it merits. This contrasts with the inclusion of risk management in Boëhm's spiral model. The traditional approach to quality relates to the term Quality Assurance (QA) which is associated with code testing. It is more appropriate to refer to *managing software quality* in order to emphasise the on-going end-to-end life cycle aspects. This, too, is illustrated in the Software Quality Star mark II by the cyclical-flow dotted line shown in Figure 4

Having identified the eight different quality perspectives (Section 2) it follows that each perspective has its own interpretation of quality. For example, when interpreting quality-of-product perspective, the topics of interest are product quality factors. Likewise, when interpreting the quality-of-procurement (ownership) and quality-of-production the topics have to do with procurement factors and production factors. Such a set of factors was identified by [14] as software quality strategic drivers. These are presented in Figure 5 where they also include familiar software quality terminology for each driver.

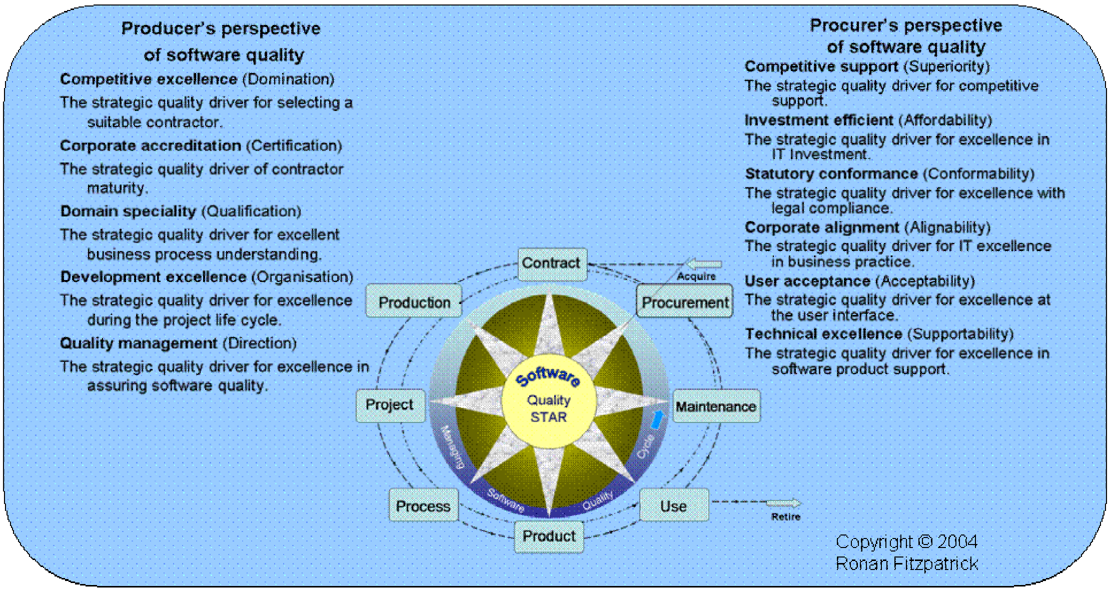


**Figure 5: Producer and Procurer strategic drivers of software quality**

## 5. Redefining quality for evolving technologies

As part of quality-of-use, ISO 9126-1 [3] explains the need to refer to context of use, that is, one product with opportunities to use it in different contexts. While the context of use may change, the domain of use is consistent - the domain is Information Systems. But the World Wide Web (WWW) is a different domain and different quality factors apply.

Multiple domains (typically the WWW) are illustrated in Figures 4 & 5 by the second cyclical-flow dotted line. Five additional quality factors for the WWW were identified by [15]. These five are *visibility, intelligibility, credibility, engagibility and differentiation* and are shown together with their sub-characteristics in Figure 6.

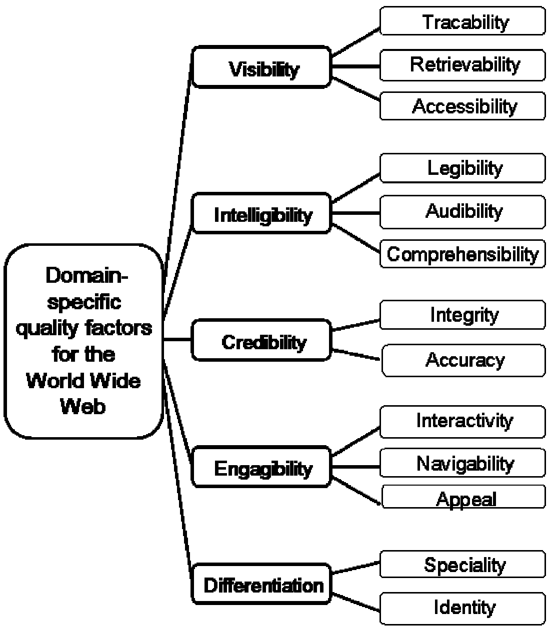


**Figure 6 - Taxonomy of domain-specific quality factors for the World Wide Web.**

The study of quality in the domain of the World Wide Web highlights new challenges as technology evolves – other domains will have different quality factors. For the WWW the challenges include methods and metrics for estimating, managing quality during the product life cycle and quality-of-use measurement. They will also include new emphasis on creating sites that support quality-of-ownership.

In their paper Software Quality Revisited [16] address challenges relating to Web site quality. They address interpreting the Strategic Drivers in relation to quality Web sites and they also address the need for measurement methods and metrics in this domain.

## 6. Conclusion

This paper has set out a number of challenges which face those interested in software quality. These include:

1. A definition of quality which focuses on measuring excellence.
2. That interpreting the term *usability* as meaning anything that impacts the end user is a more natural interpretation of the term *usability*.
3. That the broader view of quality dictates that a life cycle that focuses only on *software development* is insufficient and that a full end-to-end *software product* life cycle is required as illustrated in the Software Quality Star mark II.
4. The expression Quality Assurance does not fully address the need for quality management throughout the product life cycle.
5. New challenges are presented by the need for quality of WWW solutions.

## 7. References


[1] J. McCall, P. Richards and G. Walters (1997) "*Factors in software quality*", Vol I-III, Rome Aid Defence Centre, Italy.

[2] B. Boëhm, (1978) "*Characteristics of software quality*", Vol 1 of TRW series on software technology, North-Holland, Amsterdam, Netherlands.

[3] ISO/IEC 9126-1 (2001) *International Standard. Software engineering – Product quality – Part 1: Quality model*, International Organisation for Standardisation, Genève, Switzerland.

[4] P.B. Crosby, (1984) *Quality without tears*, McGraw-Hill books, New York, USA, p60.

[5] W.E. Deming (2000) *Out of the crises*, MIT Press, Cambridge, Mass., USA, p 168-169.

[6] A.V. Feigenbaum, (1961) *Total Quality Control: Engineering and Management*, McGraw-Hill, New York, p13.

[7] Kaoru Ishikawa, (1985) *What is Total quality control? : The Japanese way*. Prentice Hall, Englewood Cliffs, London, UK, p44/5.

[8] J.M. Juran, (1989) *Juran on Leadership for Quality*, Free Press, New York, U.S.A, p16.

[9] J.S. Oakland, (1993) *Total Quality Management*, Butterworth-Heinemann, p4.

[10] S(higeo) Shingo, (1986) *Zero Quality Control: Source inspection and the poka-yoka system*, productivity press, Cambridge, Mass, USA, page vi.

[11] Genichi Taguchi, Elsayed A. Elsayed and Thomas Hsiang, (1989) *Quality engineering in productions systems*, McGraw-Hill, New York, USA, p2/3.

[12] R. Fitzpatrick, (2003) The Software Quality Star: A conceptual model for the software quality curriculum, Workshop paper, *Proceedings of Closing the Gaps: Software Engineering and Human-Computer Interaction, INTERACT 2003*: Ninth IFIP TC 13 International Conference on Human-Computer Interaction, September 2003, Zurich, Switzerland.

[13] ISO/IEC 12207 (1995) *International Standard. Information technology - Software life cycle processes*, International Organisation for Standardisation, Genève, Switzerland

[14] R. Fitzpatrick, (2001) "Strategic Drivers of Software Quality: Beyond external and internal software quality", Second Asia-Pacific Conference on Quality Software, *Proceedings of APAQS 2001, Hong Kong*; IEEE Computer Society Press, California, USA, 2001.

[15] R. Fitzpatrick, (2000) "Additional Quality Factors for the World Wide Web", *Proceedings of the Second World Congress for Software Quality*, Yokohama, Japan, Union of Japanese Scientists and Engineers (JUSE), Tokyo, Japan.

[16] R. Fitzpatrick, P. Smith and B. O'Shea, (2004) Software Quality Revisited, *Proceedings of the Software Measurement European Forum (SMEF 2004, Rome)*, Istituto di Ricerca Internazionale S.r.l., Milan, Italy, p307/315, ISBN 88-86674-33-3.