STUDIA UNIV. BABEŞ-BOLYAI, MATHEMATICA, XL, 3, 1995

SOFTWARE QUALITY MANAGEMENT: TO UNDERSTAND AND TO INTRODUCE

I.

Ecaterina BALLA*

Received: 15.12.1994
AMS subject classification: 68N05

REZUMAT. Controlul calităţii produselor program (I). După o scurtă trecere în revistă a definiţiilor de calitate a programelor, se prezintă cel mai popular standard de calitate: ISO 9000. Sunt discutate avantajele şi dezavantajele lui. Soluţiile schiţate, bazate pe studiul şi experienţa autoarei, pot fi aplicate în toate ţările din Europa de Est, problemele de rezolvat fiind similare.

ABSTRACT. - Following a brief exposition of the history of software-quality definitions, today's most popular approach to the problem is presented: building quality management systems according to ISO 9000 Series. Some advantages and disadvantages of this trend are presented. Using results of software quality oriented research and practical training done by the author in Hungary, some problems are shown and some solutions are sketched to start introducing software quality management. The problems sketched are similar in most Eastern European countries, therefore we presume that presented solutions could be applicable as well.

Acknowledgment. Information contained in this paper is result of research and practical training supported by IQSOFT.

1. Introduction. The term quality, as "degree of excellence, relative nature or kind of character, class or grade of thing as determined by this, general excellence" has, both in philosophy and in ordinary speech, a long history. The meaning of quality has changed over

* *Technical University, Department of Mathematics and Computer Science, Budapest, Hungary, E-mail: balla@inf.bme.hu*

time according to the mentality of people, but "good quality" has always had something to do with the product's capability to satisfy the needs of the users. (Product- for some people meaning a thing, for others meaning a result of mental activity.) Requirements for quality have changed as well. For centuries quality has been mostly "a matter of conscience" and seldom an expressed need of some, wealthy groups. However, quality requirements have evolved, and this evolution has accelerated mainly in our century. The first product responsibility trial was registered in 1852 in New York (purchaser's rights had to be defended versus a company using improper inscription on chemical substances). Beginning with the 1950's many quality organizations and bodies have shown up, some of them becoming of national or international importance. Due to the work of these organizations, standards and prescriptions regarding quality have been issued, and nowadays market-position of companies highly depend on the application of these standards.

According to Armand Feigenbaum [Feigen93] world-wide market-competition has led to a change in purchasers' value judgment: they calculate in life-cycle costs, so quality comes into prominence regarded to price.

2. Quality of software. Software is "intellectual creation comprising the programs, procedures, rules and any associated documentation pertaining to the operation of a data processing system", a software product is the "complete set of computer programs, procedures and associated documentation and data designated for delivery to a user" [ISO 9000-3].
The concept of software as used today exists since the NATO conference organized in 1968 in Garmisch Partenkirchen, so it is far from being a novelty. Due to the increasing number of software users, quality is now of the same importance in developing, using and maintaining of software as in the similar activities regarding any other - material or intellectual - products. Quality of software is a concept hard to define, basically because the difficulties of finding adequate quality-criteria, attributes and proper quality-measuring methods and tools.

2.1. Basic orientations in defining the meaning of "software quality". In the "heroic age" of computer science the programs were written in most cases by mathematicians, by physicists or by engineers, and were usually concerned with the solving of some scientific

problems (numeric approach of solutions of equations, inverting a matrix etc.). These programs were dealing with simple mathematical operations that had to be performed many times. *A program was considered to be "good" if it was able to run - at least once, to terminate after a time - the length of running-time was not really important, and to produce results similar to those expected.*

In the sixties and the early seventies - mainly due to the appearance of assembly and high level programming languages and operating systems - the dimensions of the problems to be solved have increased, and there was a need for programs to perform the same operations thousands of times a day. It was the age of "tricky" programmers, quality being characterized by the so-called *"micro-efficiency". A program was said to be "good" if its use was cheap in a certain hardware-software environment, and it handled the given resources of the computer in an optimal way.*

Beginning with the eighties the concept of quality has started to mean *"macro-efficiency"*. Programs are being used in various hardware-software environments by many people among whom the percentage of "ordinary users" (not informaticians) is increasing. Due to the development of hardware there is in fact no point in "saving resources". *A "good" computer program is* therefore *portable, its code can be understood by more people, and it is highly user-friendly.*

Nowadays complex program-systems are being developed. The question isn't the correctness of a certain program any more, but the correctness, reliability, integrity, interoperability of the whole system. *A system is good if it's documentation is complete and understandable, if it interacts with other - computer aided or classic - data-processing systems in an optimal way, if it's maintenance can be done at low costs, if it is able to cooperate with new software-hardware tools* and so on.

Prescriptions for coding aren't enough any more: *complex organizational frame, well-organized team-work are needed* to insure the correct and optimal sequence of all activities performed in the development of such systems.

Beginning with the eighties *structured systems' analyzing and developing methods and methodologies* have been worked out and complex computer-aided tools have been developed, which are more then simple testing tools: they provide help for transforming the logical model

of the system into physical. Further details about such methodologies and tools are described in section 5.1.

A next step in system-development is *the use of embedded intelligence.* Programs are "living" in the electronic circuits - they become organic elements of machines. Testing such programs obviously needs techniques that completely differ from ordinary testing techniques. Working out such testing techniques and tools is a new challenge for informaticians.

2.2. Five quality definitions. Let's imagine [Genuch91] that a product has been developed according to the specification, but the users are not satisfied because the product does not fit their needs. Developers conclude that the users are not able to explain what they want, the users conclude that developers are not able to understand what the user needs. Both parties end up being unsatisfied with the result of the development effort and it is not easy to say who is right and who is wrong. In the opinion of Van Genuchten "both parties are right, according to their own definitions".

Here are the 5 quality definitions given by [Garvin84]:

- The transcendent definition says that quality is absolute and universally recognizable, despite the fact that it cannot be defined precisely. Only experience can teach to recognize quality.
- According to the user-based definition, quality is "fitness for use". This definition starts from the assumption that the individual customers have different needs and those goods that best satisfy their needs are considered to have the highest quality.
- The product-based definition views quality as a precise and measurable variable.
- Manufacturing-based definition identifies quality as conformity with specifications.
- Value-based definition defines quality in relation to costs. A quality product provides performance at an acceptable price or conformance at an acceptable cost.

The product-based definition of quality is one of the most used. This definition deals with objective quality attributes and their relationship. Quality attributes - like correctness, reliability, efficiency, integrity, usability, maintainability, testability, flexibility, portability,

reusability, interoperability - are defined, their interaction is shown by tables of correlance[2]. We can observe that using *the product-based quality definition involves some effective measurements on the software product* (e.g., number of comments in a program, number of errors, number of failures, etc.). This approach is now somehow neglected due to the new approach suggested by a popular standards series.

3. Today's fashion: ISO 9000. Among the many national and international quality organizations ISO[3] has undoubtedly reformed the concept of quality (the direction of this reform can be questionable). In March 1987 it issued the ISO 9000 Series, prepared by ISO/TC 176. In November 1987 CEN[4] adopted the set as EN 29000 Series - this was followed by a rapid adoption of the standards world-wide. The ISO 9000 News, March 1992 issue, lists 45 countries with identical adoption, and 3 countries - China, Jamaica, Venezuela - with equivalent adoptions, another booklet [Interl93] speaks about "over 90" countries that have adopted ISO 9000 by the beginning of 1993.

ISO 9000 series is a set of international standards for both *quality management and quality assurance*. ISO 8402 defines the concepts used, 9000 provides a guide for selection and use of the standards. Standards 9001 to 9003 deal with contractual situations, presenting three quality assurance models, while 9004 is in connection with non-contractual situations, presenting quality system elements used in quality management. The series has been completed with other standards, among which there is a standard for software quality management.

The ISO 9000 series was developed mainly for contractual business relationships. The goal is to increase customer confidence in the quality system used by their suppliers.

Theoretically there is no contradiction between "confidence in the quality system" and "confidence in the product's good quality", because a well designed and well maintained

---

2 Quality attributes and their relationship have been modelled in the USA for the first time, by Boehm (1977) and by McCall (1978).

3 International Standard Organisation

4 Comité Européen de Normalisation

quality system should assure the proper quality of the products.

However, there is a contradiction between the two mentioned aspects, due to the fact that *the application of ISO 9000 suggests the use of manufacturing-based definition of quality, the development process being more emphasized, while measurable quality attributes suggest the use of the product-based definition.*

One of the most important features of ISO 9000 is that it provides a third-party auditing model to review, certify and maintain certification of organizations.

3.1. The main concepts of ISO 9000. ISO 9000 Series operates with many concepts, among which some important concepts are:

• Quality = the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs[5].

• Quality policy = the overall quality intentions and direction of an organization as regards to quality, as formally expressed by top management.

• Quality management = that aspect of the overall management function that determines and implements the quality policy.

• Quality management system = the organizational structure, responsibility, activities, capabilities and resources that together aim to ensure that (software) products will satisfy stated or implied needs.

• Registration/ certification = the assessment of a company's quality system by a third party - a quality system registrar.

3.2. Registration. The company contacts such a registration body, which assesses first the documents of the quality system. In case the documents satisfy the requirements stated in the appropriate standard(s) of ISO series, an internal audit follows. The internal audit examines the compliance of the things written in the documentation with the things that really happen at the company. In most cases these reviews will expose inconsistencies that will be included in the final report. After the adequate corrective actions the auditors can fully

---

5 The following definition was suggested by [Geiger93]: **Quality** = Realised totality of characteristics and their values of an entity in relation to requirement for quality. The definition proposed at the meeting of the ISO TC 176/SC 1 (1991) was rejected with the argument that the whole world is used to taking the existing definition.

approve, conditionally or provisionally approve or disapprove the company's registration. A company is registered for a three-year period. During that time, the organization must maintain and improve the quality system that was certified.

Registration certifies the quality management system developed according to ISO 9000, not the outstanding quality of the products.

3.3. Evolution of ISO 9000. ISO 9000 Series has evolved using some earlier quality-related standards (MIL-Q-9858A, MIL-I-45208A, ANSI/ASQC A-3, DEF/STAN 05-21,22,24,25&29), among which some national standards were issued in the UK, USA, Canada, France, Germany, the Netherlands.

It is interesting to observe that Japan has not contributed with a national standard to ISO. According to [Stephens93], quality management is not invented in Japan, although many terms of the modern disciplines of quality have been borrowed from successful and innovative Japanese applications. None of these terms, concepts and techniques *represent quality system assessment, certification, registration.* In fact ISO 9000 Series was only adopted in 1991 as the JIS Z 9900 series into the Japanese national standard system, mainly dictated by the international harmony, trade and co-operation.

This shows that *quality can be produced without having a registered quality management system* - and again takes us to the idea that *ISO 9000 registration is not always the guarantee for outstanding quality.*

In opinion of many quality specialists, ISO 9000 is just the first step to Total Quality Management. It deals too much with documents, forgetting the aspects apart from assessment. A quality management system developed only in order to registrate the company cannot be viable. (See Brian Plowman's opinion in 4.2.1.)

REFERENCES

[Beiczer78] Beiczer, Ö., Szentes, J.: Felhasználói programok forráslista alapján történô minôsítése, Softtech sorozat, D26, SzKI, 1978.

[Boehm78] Boehm, B.W. Characteristics of software quality, TRW Series of Software Technology, Vol.1, North Holland, 1978

[Develin93] Develin & Partners Management Consultants, Freeing the Victims - Achieving Cultural Change Through Total Quality, 1993.

[Feigen93] Feigenbaum, A.: EOQ World Conference, Helsinki, in: MMT Társasági Tájékoztató, II. évf. 12. sz. 1993. dec.

[Garvin84] Garvin, D.A.: What does 'product quality' really mean ?, Sloan Management review, Fall 1984

[Geiger93] Geiger, W.: Talk Show with ISO 8402 ? Contemplative thoughts about the new quality vocabulary, EOQ Quality 1/1993, pg. 13-17

[Genuch91] Genuchten, M.van: Towards a Software Factory. Ph.D. thesis, Eindhoven University of Technology, 1991, ISBN: 90 900 4119 2

[Györkös94] Györkös, J.: A support for auditing the software process, in: Austrian-Hungarian Seminar on Software Engineering, Klagenfurt, 7-8 April 1994, pg. 7.1.-7.4.

[Havass93] Havass, M.: A szoftvervizsgálat múltja és jövője, II. Minőségi Hét, 1993. november 8-10., A konferencia előadásai, II. kötet, pg. 85-101

[Iterl93] The ISO 9000 Guide, Interleaf, 1993

[McCall78] McCall, J. A.: The utility of software metrics in large scale software systems development, IEEE Second software life cycle management workshop, August 1978

[Oracase] Barker, R: Case*Method, Tasks and Deliverables, Addison-Wesley Publishing Company, Revised Edition, 1990

[Pereira93] Pereira, Z. L.: EOQ World Conference, Helsinki, in: MMT Társasági Tájékoztató, II. évf. 12. sz. 1993. dec.

[PRINCE91] Project Run In Controlled Environment , LBMS Delegate Notes, 1991

[SSADM88] Downs E., Clare P., Coe I.: Structured Systems Analysis and Design Method - Application and Context Prentice Hall International Ltd, 1988)

[Stephens93] Stephens, K.: Quality Systems and Certification - Some Observations and Thoughts, EOQ Quality, 1/1993, pg 5-12

[Szentes82] SOMIKA - An automated System for Measuring Software Quality, Proc. of Premier Colloque de Génie Logiciel, AFCET, 261-276 (1982)

[Szentes83] Szentes, J.: Qualigraph - A Software Tool for Quality Metrics and Graphic Documentation, Proc. of ESA/ESTEC Software Engineering Seminar, Noordwijk, The Netherlands, 73-81, 1983.

[Szentes85] Szentes, J.: A szoftverminőség és mérése, Számítástechnika-alkalmazási Vállalat, Budapest, 1985

[TickIT90] Guide to Software Quality Management System Construction and Certification using ISO 9001/EN 29001 Part 1 (1987). Issue 1.1 (30. September 1990) British Computer Society UK Department of Trade and Industry

[ISO9000] Quality management and quality assurance standards - Guidelines for selection and use, 1987

[ISO 9001] Quality systems - Model for quality assurance in design/development, production, installation and servicing, 1987

[ISO 9004] Quality management and quality system elements - Guidelines, 1987

[ISO9000-3] Quality management and quality assurance standards - Guidelines for the application of ISO 9001 to the development, supply and maintenance of software, 1991