Quality management in software production: a customer oriented approach
Trienekens, J.J.M.

Published in:

Published: 01/01/1993

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 15. Oct. 2018
Quality management in software production, a customer oriented approach

Jos J.M. Trienekens

Graduate School of Industrial Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Abstract

Quality has been identified in the late 80s as the very core of competition in the software industry. Demonstrable high quality will become a prerequisite of survivalship for most software producing organizations. However, in software production no useful guidelines for quality management exist. In this paper we first discuss a framework for quality factors that serves as a basis of reference for quality management in software producing organizations. Then we adapt a customer oriented classification of production environments, taken from industrial production research, to the software industry. Mapping of the quality framework onto the framework of characteristics of the different software production types results in guidelines for the design of customer oriented quality management systems.

Keyword Codes: K.6.0; K.6.1; K.6.3
Keywords: Management of Computing and Information Systems, General; Project and People Management; Software Management

1. INTRODUCTION

The software industry evolved in the late 80s with the emergence of quality and measurement concepts, see Basili and Selby (1991), Eriksson and Törn (1991), Bemelmans (1991), Bush and Fenton (1990) and Kaposi and Kitchenham (1987). This new focus on software production is driven by the growing dependence of organizations on information processing and the new demands in the consumer mass market. The usual concepts of performance and costs are no longer sufficient to help determine customer strategy and many professionals, both practitioners and scientists, believe that the 90s will become the quality era of software engineering, see Basili and Musa (1991).

One of the major topics in current software quality research is quantifying software quality and bringing quality measurement to the center of software production. However, the factors that determine quality and the relations among them are not well understood and as a consequence current literature can only give some very general indications for the design of quality management systems (ISO/DIS (1991)). The main problem of such guidelines is that they are not based on particular software production characteristics of the software producing organization itself. In industrial production research it is commonly accepted that design guidelines for management and control systems have to be derived
from the customer orientation characteristics of the organization (Bertrand, Wijngaard and Wortmann (1990)). Awareness of different types of customer orientation is necessary to make a clear choice for a niche in the market place. Based on this choice the consequences for production processes, production control and subsequently quality management can be recognized.

This paper integrates quality management aspects and production control aspects of software production. In section 2 we first introduce a framework of quality factors that we will call the quality framework. Since industrialization of software production is emerging and customer orientation is becoming a topic of increasing importance in software production, see Rockwell (1991), Van Genuchten (1991) and Trienekens and Kusters (1992), we will present in section 3 a customer orientation typology of software producing organizations. Finally, we integrate in section 4 the software quality framework and the characteristics of the different types of software producing organizations in order to determine appropriate quality factors that can serve as building blocks for software quality management systems.

2. THE QUALITY FRAMEWORK: OBJECTS AND PERSPECTIVES

Quality in software production is a complex and multidimensional phenomenon. In this section we present a general framework of quality factors in software production. The framework is based on existing theories and aims at the division of factors that determine quality in accordance with the different involvements of participants in software production. The main division is in objects and perspectives.

2.1. Objects of software production

Objects are 'things' or 'artefacts' in that some party that is involved in software production is interested. A quality factor has to be interpreted as a property or a set of properties of a software production object. In fact a software quality factor is an objectively or subjectively defined function over one or more 'base' parameters, which are also called quality attributes or quality criteria, see e.g.: Kaposi and Kitchenham (1987).

In the object dimension of the framework we classify the objects in accordance with Bush and Fenton (1990) in three distinct groups: the product category, the process category and the resource category. Objects in the product category include final deliverable product. Examples of quality factors of product objects are usability, reliability and maintainability. Processes are any activities of persons involved in the production of software. Appropriate quality factors of processes are number of errors of a specified type, effort and costs. Resources are any objects used by processes excluding products of other processes. In the resource category examples of quality factors are experience, skills like communicative ability etc.

Different objects are of importance for different parties in software production. As a consequence, a particular quality factor determination of an object has to be connected with that group of individuals who have primary interest and responsibility of the rate of quality that has to be achieved, Eriksson and Törn (1991). This leads to the recognition of the dimension of perspectives.

2.2. Perspectives and approaches

In this paper we will use for each category of objects three different perspectives: the user perspective, the project manager perspective and the engineer perspective. From each of these perspectives different approaches will be used to determine the quality factors of
the different objects, and subsequently their 'base' parameters and appropriate metrics. In
figure 1 the different approaches of the distinct perspectives are shown.

In the product object category we make a distinction between respectively a user
satisfaction based approach, a system (final product specification) based approach and a
project (budget and schedule) based approach to determine quality factors. With respect to
the user satisfaction based approach examples of quality factors that can be determined are
usability, maintainability, reliability etc. With respect to the system based approach
eamples of quality factors are modularity, complexity, structuredness etc. Examples of
quality factors that can be determined in the project based approach are factors that are
results of balancing the former mentioned quality factors with time, costs and effort.

In the process object category we respectively distinguish a user participation based
approach, a design (intermediate products) based approach and a (workflow) control based
approach. With respect to the participation based approach examples of quality factors that
could be determined are participation factors such as frequency and duration of
participation. Examples of quality factors that can be determined in the design based
approach are intermediate design factors like tracebility and uniformity of design decisions.
With respect to the control based approach examples of quality factors are frequency of
verification and validation of intermediate design results.

In the resource object category we make a distinction between respectively an
interaction (resource/user communication) based approach, an allocation (resource/process)
based approach and an application (resource/product) based approach. Examples of quality
factors that could be determined by following the interaction based approach are
communicative ability, experience etc. With respect to the application based approach
examples of quality factors that might be determined are resource/design quality factors
such as the degree of formal power of methods and tools. Examples of quality factors that
could be determined by the allocation based approach are resource/workflow factors for
example with respect to putting the right persons and the right tools to the right spot.

<table>
<thead>
<tr>
<th>object perspectives</th>
<th>Product</th>
<th>Process</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>user satisfaction based</td>
<td>participation based</td>
<td>interaction based</td>
</tr>
<tr>
<td>engineer</td>
<td>system based</td>
<td>design based</td>
<td>application based</td>
</tr>
<tr>
<td>project manager</td>
<td>project based</td>
<td>control based</td>
<td>allocation based</td>
</tr>
</tbody>
</table>

Figure 1. The quality framework, objects and perspectives

Although the definitions of the approaches and the examples of the quality factors are
currently not fully satisfying, we believe that the above made clear that from each
perspective and for every object class appropriate quality factors can be identified.
It will be clear that recognition of relevant objects (and their quality factors) from relevant perspectives is not sufficient to improve quality of software production. Both measurement and evaluation are of equal importance to achieve actual quality improvement. In accordance with Basili and Selby (1991) we state that based on well defined operational goals of the involved parties, appropriate measures (subjective as well as objective) have to be determined for each of the identified quality factors of the objects. A model that shows the relation between goals and measurement is the Goal/Question/Metric (GQM)-paradigm as developed by Basili and Rombach (1988). It will be clear that goal setting, quality factor and measurement determination with respect to products, processes or resources can have completely different meanings from each of the different perspectives.

Now the quality framework has been discussed, we will present in section 3 the characteristics of three different types of software producing organizations. To establish a clear view on software production itself we will discuss in the first part of that section the main categories of activities in current software production.

3. CHARACTERISTICS OF CUSTOMER ORIENTATION IN SOFTWARE PRODUCTION

In this paper we recognize three main categories of activities which we will denote as the product modelling category, the process modelling category and the category of the reuse of previous work, see figure 2. The product modelling category is the traditional category of engineering activities, which is concerned with modelling the concepts relevant to producing a software system or the What that has to be done, e.g.: Yourdon (1989). In current software production research, more and more attention is given to the process modelling category, which is concerned with the modelling of the software production processes itself, Rockwell (1991). These modelling activities are concerned with monitoring and controlling or the How, the When and the By Whom of the software production. Of a growing importance are the activities that are concerned with the reuse of previous work in software production (e.g.: standardized components), Prieto Diaz (1991). We therefore consider managing of the reuse of previous work as the third main category of activities.
With regard to product modelling we will distinguish between four sub-functions: requirements determination, analysis, design and implementation. In process modelling there are two distinct levels of process modelling activities: the project life cycle level and the engineering activity level. With respect to the reuse of previous work we distinguish between reuse of product models (e.g.: conceptual data models, modules of code) and reuse of process models (e.g.: life cycle models, activity structures), see Figure 2.

In literature as well as in practice it is argued that software producing organizations need to pay much closer attention to other industries. They know much that software producers need to learn about production control and effective management (see Rockwell (1991)). In Trienekens and Kusters (1992), a typology of customer orientation from industrial production control research is adapted to the software industry. The main criteria for classification in this typology is the extent to which the processes are governed by the customer order. Based on the characteristics of the activities in the primary software production processes three types of software production were defined that we will call in this paper respectively: engineer-from-components, engineer-from-products and engineer-from-scratch (see Wortmann (1991)). Each of the three software production types has different characteristics with respect to product modelling, process modelling and reuse of previous work, see Figure 3. We will explicate these characteristics in the following.

In engineer-from-components product variety is restricted to a number of preselected families of products. Products or their components are market oriented developed instead of customer oriented developed. The emphasis in engineer-from-components software producers is therefore on the lower life cycle phases and on the reuse of both formal product components (e.g.: modules of code) and formal process components (e.g.: workflow activity structures). With respect to process modelling there are hardly any variations in the software production process on the life cycle level. Therefore the emphasis will be on the modelling of activity structures, see figure 3.

In the engineer-from-products type, software production is focussed on specialization in specific application areas. Requirements of the client of the software product are restricted to a limited product range; emphasis in product modelling will be on analysis, design and implementation. In the requirements and design phase of the life cycle, previous informal product and process models will be used as an aid of reference. Due to the moderate
customer dependency in this type of software production, movement down the life cycle will not necessarily be linear. Instead, various life cycle models will be developed dynamically, reusing life cycle models from previous production situations, see figure 3.

In engineer-from-scratch software production there is no specialization in typical products, purely production capacity is sold to the customer. In this type of software production software producers are used to operate with different product modelling approaches in all life cycle phases and the product models and life cycle models models will differ with respect to formality, completeness, consistency etc. As a consequence there are hardly any possibilities to reuse previous work, see figure 3.

Figure 3. Characteristics of the different types of software production

<table>
<thead>
<tr>
<th>PRODUCT MODELLING</th>
<th>ENGINEER FROM COMPONENTS</th>
<th>ENGINEER FROM PRODUCTS</th>
<th>ENGINEER FROM SCRATCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESIGN AND</td>
<td>ANALYSIS, DESIGN, AND</td>
<td>REQUIREMENTS,</td>
<td>MANAGING THE REUSE OF PREVIOUS WORK</td>
</tr>
<tr>
<td>IMPLEMENTATION</td>
<td>IMPLEMENTATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPECIFICATION</td>
<td>SPECIFICATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORMAL PRODUCT</td>
<td>INFORMAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPONENTS</td>
<td>PRODUCT REFERENCE MODELS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FORMAL ACTIVITY</td>
<td>INFORMAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRUCTURES</td>
<td>LIFE CYCLE REFERENCE MODELS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTIVITY MODELLING</td>
<td>LIFECYCLE MODELLING</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIFECYCLE MODELLING</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. Characteristics of the different types of software production

4. QUALITY MANAGEMENT IN CUSTOMER ORIENTED SOFTWARE PRODUCTION

This section uses the quality framework (section 2, Figure 1), and the characteristics of the different software production types (section 3, Figure 3), to integrate the quality management aspect and the production control aspect in software production. The production terminology that we used to express the results has partly been derived from (Wortmann 1991). We restrict ourselves in this paper to the highlights of the results of integration.

- Quality management in engineer-from-scratch software production. Important characteristics in this type of software production, i.e. the strong customer orientation, the absence of predefined products and the lack of formal specified processes, point in the direction of an interaction based approach to determine quality factors of (human) resources, e.g.: skills and experience of developers to stimulate cooperation.
So we have indications that quality management in this type of software producing organization will have a strong capability orientation.

- Quality management in engineer-from-products software production.
Characteristics of this type of software production, for example the moderate customer orientation and especially the use of (global life cycle) process models and product models as a reference, point in the direction of both process and product oriented quality management. However, since software production processes are hard to define in this type of software producing organization, see: Humphrey (1988), product quality management might be the primary objective for management. Main indications for quality management could be derived from both a user satisfaction and a system based approach of the determination of product quality factors.

- Quality management in engineer-from-components software production.
In this type of software producing organization, components with predefined quality attributes are reused. Here, the defining characteristic is the ability to monitor and control the production process. Although component quality is an important issue, the main quality management instrument will be process control. For example the measurement and evaluation of the structure of complex sets of tasks (see: Rockwell (1991)). The development of process models and the reuse of previous work, e.g. standardized components, will be reflected by a quality management that is based on the determination of process quality factors. So we have indications that quality management in this type of software producing organizations will have a strong process orientation.

It should be said here that the mapping is simplified and that only some of the highlights of the results are given. It will be evident that in each of the three types of software production also other quality factors, from other perspectives and for other objects, can be recognized. The objective in this paper was however, also for the sake of clarity, to emphasize only the most important quality factors.

5. CONCLUSIONS
This paper presents a novel way to derive guidelines for the design of quality management systems in software producing organizations. Guideline derivation is based on the integration of two frameworks that are constructed from respectively the viewpoint of software quality management aspects and the viewpoint of software production control aspects. From the viewpoint of software quality management, an integral measure oriented quality framework is presented that can be used as a reference basis for the determination of quality factors. From the viewpoint of customer oriented production control, we presented important differences in the characteristics of three types of software producing organizations. The integration results show some appealing mappings between software quality factors and the type of software production.

The three types of software producing organizations presented are ideal types, i.e. extremes chosen for the sake of clarifying the concepts introduced. However, it is our contention that, as in other industries, software producing organizations will have to make clear choices with respect to the niche in the market place that the organization is aiming at in order to gain competitive advantage. The recognition of the consequences of this aim for the software production processes will provide useful insights in the determination of
the relevant quality factors and subsequently their metrics and will serve as a bases for the
derivation of guidelines for the design of software quality management systems.
Currently the evidence of the developed constructs is mainly intuitive. Further steps in
this research are aimed at the validation of the proposed integration and the achieved
design guidelines for software quality management systems.

6. ACKNOWLEDGEMENTS

Finally the author would like to thank Hans Wortmann and Henk Jan Pels for their
constructive criticism and their helpful comments that resulted in the final version of this
paper.

7. LITERATURE

Basili V.R., J.D. Musa, "The future engineering of software: a management perspective"
Basili V.R., R.W. Selby, "Paradigms for experimentation and empirical studies in software
engineering". In: B. Littlewood and D. Miller (eds.), Reliability Engineering and
Bemelmans T.M.A., "Management Information Systems and Automation". The
Bertrand J.W.M., J. Wijngaard, J.C. Wortmann, "Production Control, a Structural and
152-157.
Hum prey W., "Characterizing the Software Process: A Maturity Framework". IEEE Soft-
ware, 1988, pp. 73-79.
ISO/DIS 9000-3, "Guidelines for the application of ISO 9001 to the development,
supply and maintenance of software", Quality Management and quality assurance
Kaposi A., B. Kitchenham, "The architecture of system quality". Software Engineering
Pietro Diaz, R. "Implementing Faceted Classification for Software reuse". Communications
Rockwell R., "Software Factories: The industrial production of mind stuff"
In: B.E. Hirsch, K.D. Thoben (eds.), New Approaches towards 'One-of-a-Kind'
Trienekens J.J.M., R.J. Kusters, "Customer Orientation as a Basis for Computer Supported
Technologies in Software Production". In: K.E. Kendall, K. Lyytinen, J.I.DeGross
(eds.), Proceedings of the IFIP 8.2 Working Conference on The Impact of Computer
Supported Technologies on Information Systems Development. Amsterdam, New York: