

Customer oriented design of quality management systems (QMS) : learning from industrial manufacturing

Citation for published version (APA):

Trienekens, J. J. M., & Thoma, P. H. C. M. (1994). Customer oriented design of quality management systems (QMS) : learning from industrial manufacturing. In M. Ross, C. A. Brebbia, & G. Staples (Eds.), *Software quality management II. Vol. 2. Building quality into software* (pp. 18-32). Computational Mechanics Publications.

Document status and date:

Published: 01/01/1994

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 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](#)

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.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Customer oriented design of quality management systems (QMS): learning from industrial manufacturing

J.J.M. Trienekens & P.H.C.M. Thoma

Graduate School of Industrial Engineering and Management Science, University of Technology Eindhoven, The Netherlands

ABSTRACT

Software producing organisations aim at a more industrial production of software. They need to identify their niche in the market place and they have to improve quality of both products and processes. In this paper a new approach is presented to derive guidelines for the design of quality management systems (QMS) in software producing organisations. This approach is based on lessons that the software industry can learn from industrial manufacturing. A typology is introduced that distinguishes between engineering-from-scratch, engineering-from-references and engineering-from-components software production. This typology has been applicated in the analysis of the business processes of two different software producing organisations in The Netherlands. Both organisations struggle with questions about how to specify their products, how to define their production processes and how to improve quality. From the production control characteristics of the two organisations guidelines are derived for the design of software quality management systems.

1. INTRODUCTION

The software industry emerged in the late 1980s with quality and measurement concepts (eg. Basili and Musa [3]). Quality in particular is becoming the core of competition in the software industry. Therefore many software producing organisations currently focus on the development of a quality management system (QMS). The importance of a QMS can be considered from two perspectives. On the one hand a software producing organisation can improve their own development and engineering process by defining and structuring their activities and procedures. On the other hand, by developing

section 3 we describe the research concepts that we used. Section 4 presents the main results of our research. We will report on both the characteristics of control aspects and operational processes of the two organisations. Based on these characteristics the ISO 9000-3 framework is used to point out the main QMS elements for the two different organisations. In addition we report on the role and meaning of the reuse of previous work with respect to the evolution of software quality management systems. Section 5 summarizes this paper.

2. SOFTWARE PRODUCING ORGANISATIONS IN THE NETHERLANDS

The organisations that were subject of our research are two important software producers in the Netherlands, respectively Baan International and Triton Professional Services. Baan International (BI) is a software producing organisation that is specialized in the development and international distribution of business control information systems which are offered to the market as standard software products called Triton. Triton Professional services (TPS) uses Triton as a basis for customization (eg. adds new software components) and produces client-specific products. Triton is a family of more than thirty UNIX-based application modules. Generic standard applications are created by Baan International (without intervention of TPS) by selecting and assembling appropriate modules. The applications serve different markets, being: Manufacturing, Wholesale, Finance and Service & Maintenance.

It will be clear that Baan International and Triton Professional Services have different customer orientation characteristics. Baan International can be characterized as an engineer-from-components organisation. Although Triton Professional Services has some engineer-from-scratch activities its main focus is on engineer-from-references (the standard product Triton is used as a reference basis). Although the fast international expansion of Baan International (with services in more than 30 countries) and Triton Professional Services led to very satisfying business results, it also caused various problems regarding quality. Problems which also affect organisations who use BI's generic standard products as a bases for customer dependent development, such as TPS. Both organisations decided recently to spend a considerable amount of their effort on the design of a software quality management system. Before we will report on the results of our research we shortly present the research concepts that we used.

3. RESEARCH CONCEPTS

The main research concepts that we used in our research are respectively: the notion of different product structures (Bertrand et al [5]), the distinction between three categories of software production activities (Trienekens and

Kusters [15]) and the process-control paradigm (Bemelmans [4], De Leeuw [7]).

3.1 Different types of product structures

In industrial manufacturing literature it is argued that the position of the customer order decoupling point reflects important production and control characteristics of an organisation. The customer orientation is related to the extent to which the production processes are governed by the customer order (Sari [12]). We call this the customer dependency of the production process.

Traditionally two types of dependency can be distinguished. Customers demanding fast delivery are supplied with standardized products and customers with specific demands are presented with products designed to order. It will be clear that problems encounter if a customer demands a specific product to be delivered with a short lead-time. In order to satisfy this demand, a restructuring of the production process is necessary. Instead of the production of unique products by assembling a large number of specific components an organisation strives at the assembly of a range of customer specific products on the basis of a limited number of standardized components.

This shift is represented in Figure 1, where the isolated pyramids represent the traditional product structures, with components specific to the product reflected by their bottoms and the (unique) client-specific end-products by their tops.

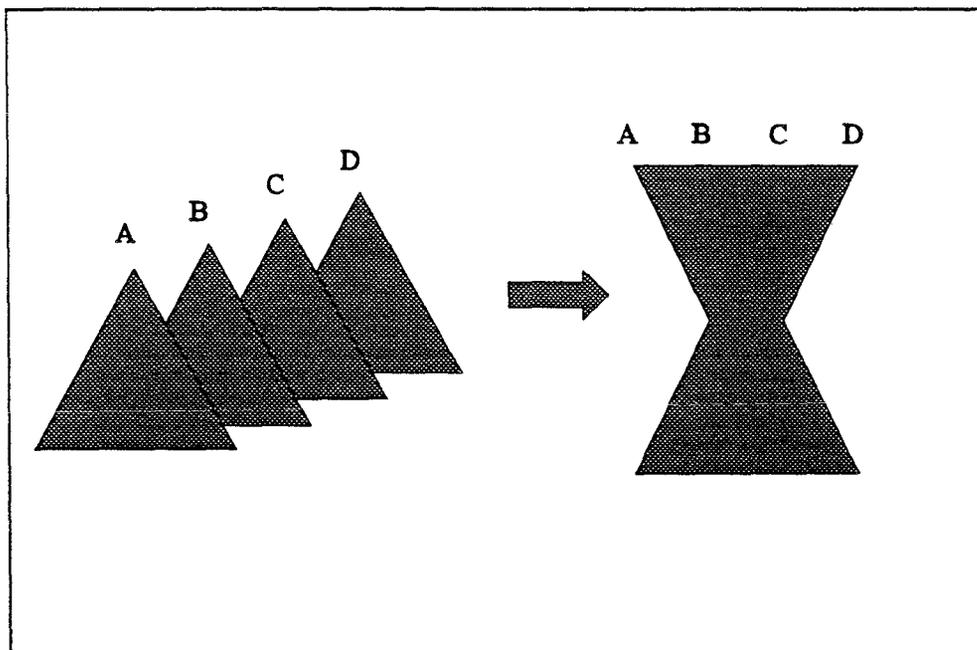


Figure 1: Different product structures

The hour-glass shows the notion of assembling a variety of end products using a limited number of components. The limited number of components are reflected by the neck of the hour-glass and the larger number of client-specific end-products by its top. Key to the success of such an approach is the explicit reuse of standardized components enabling a reduction of leadtime (for the customer) to be coupled with a client-specific design of the final product (van Genuchten [16]).

The software industry markets puts pressure on software producing organisations to choose their own niche within the possibilities set out above. There are possibilities for organisations specialized in the fast delivery of relatively inexpensive standard products, but also for organisations who design expensive and complex customer specific products.

3.2 Three categories of software production activities

In software production, different related activities have to be distinguished (eg. Basili and Rombach [1]). In this paper we use a distinction between three main categories of activities which we will denote as the product modelling category, the process control category and the category of managing the reuse of previous work (Trienekens and Kusters [15]).

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 (eg. Yourdon [17]). Software products are always developed by modelling the product in a number of life cycle phases (eg. requirements, design, construction, implementation). In each phase the particular product developer (requirements analyst, designer, constructor) has to deal with different specific development constraints (from type organisational, conceptual, technical).

In current software production research, more and more attention is given to the process control category (Rockwell [11]). Producing software is more than producing a collection of product models. The flow of work that has to be done by the various players in the production of software has to be structured and controlled. These control activities are concerned with the How, the When and the By Whom of the software production, in terms of roles, tasks, allocations and resources.

In the software industry there is a growing importance of the activities that are concerned with the reuse of previous work (Biggerstaff and Perlis [6], Prieto-Diaz [10]). During the life cycle many different products of the system being developed will be delivered in several versions and configurations. The reuse of this variety of previous work has to be managed. Although reuse of processes is of equal importance as the reuse of products we restricted ourselves in this research to two types of reusable elements,

22 Software Quality Management

respectively: informal product references (eg. domain analysis models, requirements specifications etcetera) and formal product components (eg. design specifications, interface descriptions, modules of code etcetera) (Trienekens and Kusters [15]).

3.3 The process-control paradigm

To gain a deeper understanding of the problems that both organisations encountered we used the process-control paradigm as an analysis instrument (Bemelmans [4], De Leeuw [7]).

The process-control paradigm distinguishes between the operational activities (i.e. modelling or engineering) and the control activities (i.e. project management) in a business situation. In accordance with the process-control paradigm the operational activities have to be analyzed from a system science point of view. This means that a distinguishment has to be made between the input, the transformation, the output and the resources (both people, methods and tools). Aspects of the control activities that we distinguished are respectively the type of project organisation, the type of order management, the documentation phases, the acceptance procedures and the maintenance policy. Between the operational activities and the control activities interrelations are defined.

By means of explorative interviews and analysis of project and design documents the characteristics of both organisations were investigated. We used structured in-depth interview techniques to interview persons in both organisations. Among the persons that we interviewed there were representatives from general management, project management, and software engineering.

4. MAIN RESULTS OF THE RESEARCH PROJECT

In this section we will present the characteristics of the two software producing organisations that we identified. Starting point is the notion of different types of product structures. We will see that different product structures ask for different production control concepts, different product modelling aspects and different control aspects. Consequently guidelines will be derived for the situational design of software quality management systems.

4.1 Characteristics of the software producing organisation

Different production control concepts at BI and TPS By using the hour-glass of product structures that we introduced in section 2 we were able to describe the different positions of the customer order decoupling points of the organisations.

For Baan International the position of the customer order decoupling point is situated at the end of their software production activities. The products are generic standard products (see Figure 2). The customer dependency is low because of the market orientation of the development process. Development projects are carried out without intervention of customers. Customers can express their wishes to be heard but development is not initiated on their order and their wishes do not have to be fulfilled. This production control concept is referred to by the name engineer-from-components or make-to-stock.

Triton packages cannot be bought at Baan International. The software products are distributed via dealers. Using the limited number of standard products of BI the dealers assemble and configure a larger variety of customer specific products. A shift takes place from generic to specific (see Figure 2). BI and its dealers together can be referred to by the production control concept engineer-from-components.

The interviews showed that for approximately 80% of the customers additional customization activities are necessary. Triton Professional Services uses the customer specific products of the dealers of Baan International as a reference base for customization activities (see Figure 2). The production control concept of TPS, where development takes place on the basis of a specific assignment and where the requirements of a client are restricted to a limited product range, is called engineer-from-references.

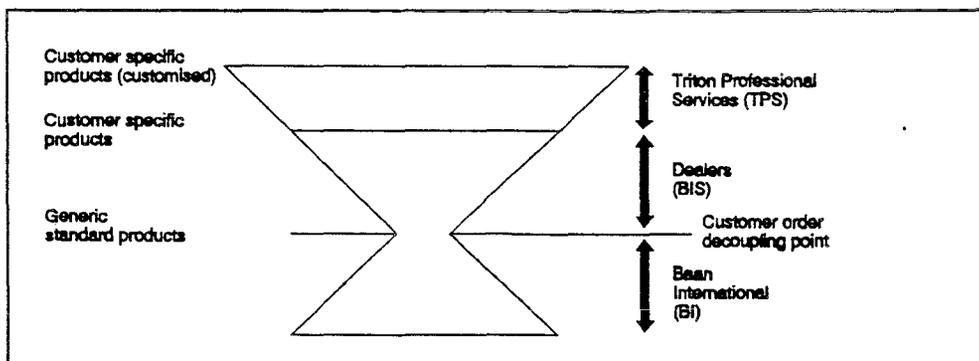


Figure 2: Different customer order decoupling points for different organisations

24 Software Quality Management

Different modelling aspects at BI and TPS By applying the process-control paradigm to analyse the business process of Baan International and Triton Professional Services we were able to identify main characteristics of both the modelling activities and the control activities. We will describe them in short.

With respect to the modelling part of software development we investigated the input (i.e. the way software requirements are identified and defined), the process (i.e. the way software products are developed), the output (i.e. the amount in that the products meet the customer requirements), the type of participants that are involved and the methods and tools that are used. Highlights of the characteristics of the engineering processes of Baan International and Triton Professional Services are respectively:

Baan International:

- the input: the organisation determines the type of products that have to be made (based on market information)
- the modelling process: the organisation develops successive versions of products of one and the same product family
- the output: many versions in past years; however there are quite some 'calls' from dealers with respect to faults in the software products of their customers
- the participants: many employees of the organisation are involved in the development processes; for testing activities also employees from dealers are invoked

Triton Professional Services:

- the input: the customer decides what functionalities have to be built into (or add to) the product; some restrictions are in force, i.e. there must be a possibility to develop the functionalities with Triton as a basis; the customer determines specific products that have to be made
- the process: a new customer specific product is made by way of modifying the standard products of BI or by adding new functionalities to them
- the output: quite often difficulties arise with the acceptance of the product because of differences between the product the customer had in mind and the product that has been developed
- the participants: the customer often participates very intensively in the process; in most cases only one employee of TPS is taking part in a specific project

With respect to analysis and design methods and tools we found no differences between the two organisations. On an organisational management level it was decided to use the same methods and tools because of the same Triton products both organisations dealt with. At the time of our research the

organisations were not aware of the need for different methods and tools because of the differences in their business processes.

Different control aspects at BI and TPS With respect to the control (management) part of software development we investigated the type of project organisation, the management of orders, the documentation phases that were distinguished, the acceptance procedures and the maintenance policy. Highlights of the characteristics of the control processes of the two organisations of Baan International and Triton Professional Services are respectively:

Baan International:

- project organisation: the projects are executed on the bases of largely existing organisation structures, projects are nearly always long-term projects and many employees are involved (about: 60 persons).
- order management: does not exist
- documentation phases: are always identical for each of the projects, it is always the management that decides on starting new phases
- acceptance: of a new (version of a) software product is speeded up by early involvement of dealers
- maintenance: problem-solving takes place per version of each type of software product

Triton Professional Services:

- project organisation: short projects, often only one expert that executes the project
- order management: difficult and time-consuming contract development
- documentation phases: the client has a substantial influence on the sequence of the processes that are carried out
- acceptance: is always based on agreed functionality
- maintenance: isolated projects per customer

To get a clear picture of the main differences of the two software producing organisations the identified characteristics are summarized in Table 1.

From that Table it can be concluded that different types of organisations (with different customers and different products) ask for different production control concepts, different modelling aspects and different control -aspects. Consequently we will use these investigations to derive guidelines for the design of software quality management systems. In the next section we will use the ISO 9000-3 quality framework to point out some differences of software quality management systems for the two organisations we previously discussed.

	<i>Baan International</i>	<i>Triton Professional Services</i>
Production Control Concept		
	Engineer-from-components - management initiates process - BI determines functionality	Engineer-to-references - customer initiates process - customer and dealer determine functionality
Modelling		
<i>Products</i>	- development of versions of product-types - BI determines type of application	- customization of standard products - customer determines specific attributes
Control		
<i>Project Organization</i>	- largely existing organisation structures - long-term projects - many employees involved	- project organization - short-term projects - few employees involved
<i>Order Management Phases/Activities</i>	- none - identical for each project - management decides on the start of new phases	- contracts - customers influence phases and activities - new phases after validation and verification of customer
<i>Acceptance</i>	- based on involvement of dealers	- based on agreed functionality and performance
<i>Maintenance</i>	- problem-solving per version	- problem-solving per customer

Table 1: Characteristics of two software producing organisations

4.2 Situational design of software quality management systems

The ISO 9000-3 framework The ISO 9000-3 standards provide guidance to facilitate the development and the certification of software quality management systems. Table 2 shows a breakdown structure of the different elements of a software quality management system according to ISO 9000-3. Three main categories of guidelines are distinguished: the quality system framework, the life cycle activities and the supporting activities.

ISO 9000-3 Breakdown		
Quality System Framework	Life Cycle Activities	Supporting Activities
Management Responsibility of the Quality System	Contract Review	Configuration Management
Internal Quality System	Purchaser's Requirements Specification	Document Control
Audit	Development Planning	Quality Records
Corrective Action	Quality Planning	Measurements
	Design and Implementation	Rules, Practices and Conventions
	Testing and Validation	Tools and Techniques
	Acceptance	Purchasing
	Maintenance	Included Software Product
		Training

Table 2: the ISO 9000-3 framework

ISO 9000-3 doesn't distinguish between different types of software producing organisations. By acknowledging different types of software producing organisations and based on their software production characteristics we pointed out the different ways that the ISO 9000-3 elements should be addressed. For each of the ISO 9000-3 framework categories we will give examples of interpretations and applications of standards.

Baan International, Triton Professional Services and the applicability of ISO 9000-3 With respect to the guidelines in the category *quality system framework* BI and TPS have different notions of management responsibility, respectively responsibility for markets and responsibility for customers.

With respect to the guidelines of the *life cycle activities* category BI does not need procedures for customers for *contract review*. For TPS on the contrary they are very important. *Acceptance* for BI means in the first place that dealers have to give their approval. The dealers' opinion of the product can be influenced by early involvement and by supplying them with a complete and consistent set of documentation and knowledge about the product. For TPS a good requirements specification and contact with the customer on a regular basis smooth the acceptance.

Other life cycle activities are executed in totally different ways. Because of the strong client orientation, for TPS the obtaining of an *unambiguous, complete and consistent list of requirements* and *the identification of the initial status* of the product are essential. It serves as a basis for develop-

28 Software Quality Management

ment, acceptance and order management.

Considering the fact that development projects are always planned in the same way (phases/activities are identical for each project) and considering the long duration of projects and the number of employees and departments being involved, BI is able to *develop plans that determine and monitor the process quality on a detailed level*. The quality of the software products depends on the mutual cooperation among product groups and departments.

For TPS the situation is different. Here only general life cycle models can be used. There is hardly any communication between the specialists that are doing the work.

With respect to the guidelines of the *supporting activities* category we point out the necessity of *configuration management* and *documentation control* for BI (especially an up-to-date functional design is important). This is stated due to the fact that the software products that are made during each phase are used for a go/no-go decision for the next phase and the fact that BI makes successive versions of one product.

For TPS the skills, experience and abilities of the software engineers play a key role in obtaining the correct specifications; *training* of software engineers plays an important role.

This paragraph presented some key differences in quality management system elements of the two organizations based on an analysis of their production control characteristics. In the next section we will focus on the evolution of the software quality management systems.

4.3 Evolution of software quality management systems: reuse of previous work

An efficiently operating quality management system cannot be established and implemented in one go. A quality management system has to evolve over time. In our research, evolution of a quality management system has been focused on the reuse of previous work.

As mentioned before (in section 3.2), we distinguished in our research, two types of reusable elements: product references and product components. The results of the analysis showed that the different characteristics of BI and TPS also influence the type of previous work that is appropriate for reuse.

Reuse of previous work at Triton Professional Services (TPS) For TPS we looked at the possibilities to reuse customizations of a specific client for another client. As opposed to reuse of references, reuse of code was not found to be attractive. The reason is the strong customer orientation; the code that is made for a particular client is too specific to be used for another client without changing it. In most cases it will be easier (i.e. less time-consuming and less expensive) to engineer these applications from scratch.

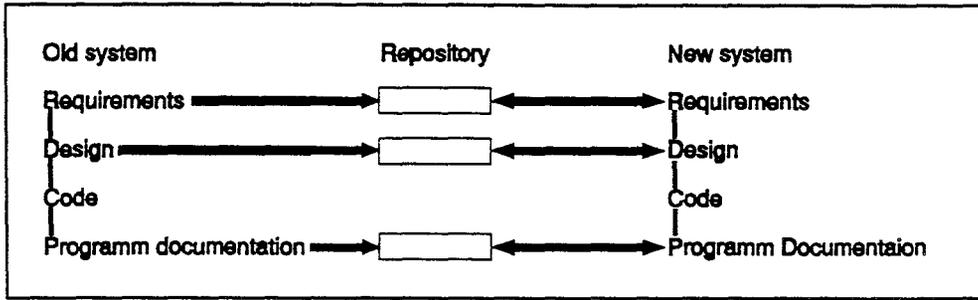


Figure 3: reuse of references at Triton Professional Services (TPS)

Figure 3, that is derived from (Basili [2]), shows that TPS should pursue reuse of references and should not try to reuse code. New developments start with the requirements and the design of the new system making use of requirements, design and program documentation of previous systems.

The main goal of reuse of references in the TPS situation is to improve the relationships with their clients. Quality is defined as user based quality (i.e. 'fitness for use'). Achieving quality has to be based on knowledge of the wishes and the needs of the customers and on insight in their specific business situations. In a software quality management system one of the key elements for continuous improvement will be a repository of reference documents which defines previous systems and project experiences and in which reference documents of new systems can be stored for future development.

Reuse of previous work at Baan International (BI) For BI we looked at the possibilities to reuse products of earlier versions for the development of new modules (being part of a new version). Research showed that especially reuse of code elements (in combination with their functional specifications) should be pursued. In order to be suitable for reuse the code elements should not be too specific (eg. too small). In the business situation of BI, functions (comparable with functions in C or Pascal) were pointed out as appropriate elements.

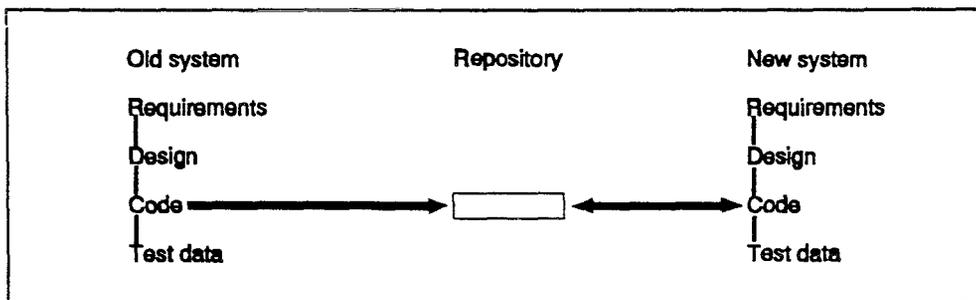


Figure 4: reuse of code at Baan International

30 Software Quality Management

Figure 4 shows the reuse orientation of BI (derived from Basili [2]). For the development of a new version, existing functions are being used. The main goal of reuse of code components in the BI situation is the improvement of the structure and the configuration aspects of their software products. Maintainability has to be improved on the bases of standardization and modularization of software code components. Quality is defined as product based or manufacturing based quality (i.e. 'conformance to specification').

In a software quality management system one of the key elements for continuous improvement is a central library and an organisational unit that is authorized to prevent and to correct faults in code components, to strive at code reduction, to improve the standardization of code etcetera.

Because of the high expectations of Baan International of reuse of code, the reuse policy for BI in the context of the QMS has been elaborated further.

Reuse and the evolution of the Quality Management System at Baan International Evaluation of the business situation of BI, in which procedures for reuse of functions had recently been set up, learned us that reuse didn't seem to get off to a good start. The main reason for this were conflicting interests; software engineers did not have any time to develop reusable functions within project constraints and as a result the functions that were made by them were not suitable for reuse.

Propositions for organisational measurements have been made to improve that situation. Starting with an investigation of currently available functions and by means of training and educating software engineers, a situation is realised in which there is a separation of concerns between design of products and design of reusable components.

Figure 5 shows the consequences for BI. The recommendations that we formulated were the following. In the next years two separate and parallel engineering activities will have to take place:

- all reusable components are developed by a separate functional components design organisation. The neck of the little hourglass represents the standardised functional components being designed by this organisation;
- software engineers (designers) use the standardised functional components to configure and assemble new generic standard products; they also develop the non-reusable parts of the generic standard products.

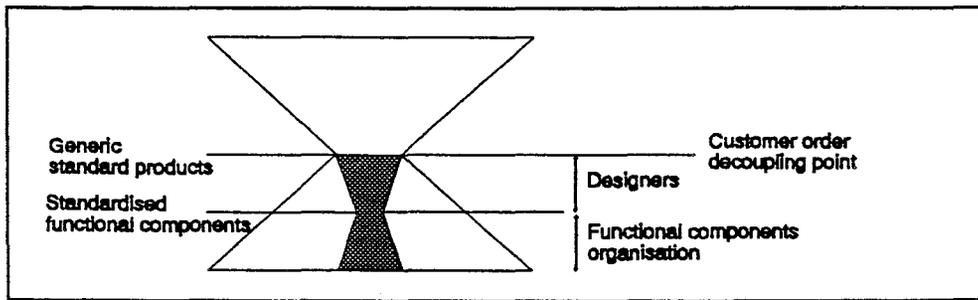


Figure 5: Development and assembly in engineer-from-components software production

Baan International considered these recommendations as important with respect to the development and evolution of a software quality management system. The organizational measures that had to be taken to facilitate the reuse of software components are considered as fundamental for their business. As a consequence standards and measures for the different types of software components have to be developed and have to act as key elements in the evolution of their software quality management system.

5. SUMMARY AND CONCLUSIONS

The main message of this paper is that production management concepts from industrial manufacturing offer new challenges to the software industry. This paper focused on the derivation of guidelines for the design of quality management systems.

The development of different software products, i.e. for markets and for individual customers, asks for different production control concepts. These differences were investigated in this paper for two different software producing organisations, i.e. an engineer-from-references and an engineer-from-components organisation. We identified respectively the differences in production control concepts (in this paper we focused especially on the customer order decoupling point) and the differences in modelling and control activities. Based on these investigations we adapted the ISO 9000-3 framework to the two different software producing organisations. From these adaptations, guidelines were derived for the design of software quality management systems.

In addition we considered the evolution of a quality management system for the two organisations at hand. In the engineer-from-references organisation the ultimate goal is to improve the relationships with clients. The business orientation on quality is a user based one. In an engineer-from-components organisation the main objective is to improve the standardization and the modularisation of software products. Here the business orientation on quality is a product and a manufacturing based one. These two different views on reuse ask for different quality standards and measures. In the two

organisations that we discussed in this paper we found that learning from previous work, i.e. reuse, had to be considered as a key factor with respect to the evolution of a software quality management system.

REFERENCES

1. Basili V.R. and H.D. Rombach, 'The Tame Project: Towards improvement oriented software environments' *IEEE Transactions on Software Engineering*, Vol.14, pp. 758-773, 1988
2. Basili, V.R., 'Viewing Maintenance as a Reuse-Oriented Software Development' *IEEE Software*, Vol.7, pp. 19-25, 1990
3. Basili, V.R. and J.D. Musa, 'The future engineering of software: a management perspective' *IEEE Computer*, Vol.2, pp. 91 - 96, 1991
4. Bemelmans, T.M.A. *Business Information Systems and Automation* (in Dutch) Stenfort Kroese, Leiden and Antwerpen, 1991.
5. Bertrand, J.W.M., Wijngaard, J. and Wortmann J.C. *Production Control, a Structural and Design Oriented Approach* Elsevier, Amsterdam, 1990.
6. Biggerstaff T.J., A.J. Perlis, *Software Reusability, Vol. 1: Concepts and Models, Vol. 2 Applications and Experience*, Addison-Wesley, 1989.
7. De Leeuw, A.C.J. 'The control paradigm as an aid for understanding and designing organisations,' in *Progress in cybernetics and systems research*, Hemisphere Publishing Corporation, London, 1979.
8. Humphrey, W. 'Characterizing the Software Process: A Maturity Framework' *IEEE Software*, Vol.5, pp. 73 - 79, 1988
9. ISO 9000-3, Quality management and quality assurance standards, Part 3: Guidelines for the application of ISO 9001 to the development, supply and maintenance of software, *International Organization for Standardization*, 1991.
10. Prieto-Diaz, P. 'Status report: software reusability' *IEEE Software*, Vol.10, pp. 61 - 66, 1993
11. Rockwell, R. 'Software Factories: the industrial production of mind stuff,' in (Ed. Hirsch, B.E., Thoben, K.D.), pp. 195-213, *Proceedings of the International Working Conference on New Approaches towards 'One-of-a-Kind' Production*, Bremen, Germany, 1991.
12. Sari, J.F. *The MPS and the Bill of Material go hand-in-hand* Richard C. Ling Inc., New York, 1981.
13. Thoma, P. 'Give way to quality' (in Dutch: 'ruim Baan voor kwaliteit') *Internal report University of Technology Eindhoven*, Eindhoven, The Netherlands, 1993.
14. Trienekens, J.J.M. 'Quality Management in Software Production, A Customer Oriented Approach,' in *Computer Applications in Technology* (Ed. Pels, H.J. and Wortmann, J.C.), pp. 127-137, *Proceedings of the IFIP 5.7 Working Conference on Integration in Production Management Systems*, Eindhoven, Nederland, 1992. North-Holland, Amsterdam, London and New York, 1992.
15. Trienekens, J.J.M. and Kusters R.J. 'Customer Orientation as a Basis for Computer Supported Technologies in Software Production,' in *Information Systems* (Ed. Kendall, K.E. and Lyytinen. K.), pp. 315-331, *Proceedings of the IFIP 8.2 Working Conference on The Impact of Computer Supported Technologies on Information Systems Development*, Minneapolis, Minnesota, USA, 1992. North-Holland, Amsterdam, London and New York, 1992.
16. Van Genuchten, M. *Towards a software factory* Kluwer Academic Publishers, London, 1992.
17. Yourdon, E. *Modern Structured Analysis* Yourdon Press, New York, 1989.