Customer-Oriented Planning of CASE-Tools Using Quality Function Deployment (QFD)*

Georg Herzwurm, Werner Mellis, Dirk Stelzer

Chair of Business Computing, University of Cologne, Prof. Dr. Mellis
Albertus-Magnus-Platz, 50923 Cologne, Germany

Abstract

Software products often do not satisfy customer expectations. This is especially true for CASE tools. Much of the success of Japanese companies particularly in meeting customer expectations is commonly attributed to their employment of simultaneous engineering. Simultaneous engineering uses a synchronous definition of product-design and production-processes based on customer’s requests and interdisciplinary team work in the company. QFD is the backbone of simultaneous product-planning. OFD was originally developed for industrial production. This paper describes a case study of applying QFD to the planning of a CASE-Tool and demonstrates how to use and adapt the QFD method for application to software products.

1. Customer-Orientation of CASE Tool Producers

According to Gartner-Group only ten European companies are among the world’s hundred largest software-houses. The CASE-Tool market is dominated by American producers, too. There has not been much research about the reasons for this dominance. The Irish Software Department for example mentions the following reasons: insufficient marketing and a lack of customer-orientation in Europe.

Research at the University of Cologne proves, that CASE tool providers are often more technology oriented than customer-oriented. They strive to offer newest technology and elaborate functionality, while customers give highest priority to non functional requirements like user friendliness and adaptability. [1] Frequently CASE tool providers seem to be engaged in missionary activities as in the following example. The client demanded the capability to print out an entity relationship diagram distributed on several sheets of paper. This request was rejected by the CASE tool producer with the argument, that a „good“ entity relationship diagram should fit onto one sheet of paper. This and similar examples demonstrate that the discrepancies between customer expectations and product characteristics rely in a lack of customer-orientation. Often the tool is a laboratory invention, which was never confronted with a systematic analysis of customer needs and wants or the known customer requirements have not been properly deployed.

2. Basic Principles of CASE Tool Planning

In the end it is up to the customer to decide about success or failure of a product. Where the determining factor is the customer’s perception of the value the product creates in his application. For a CASE tool this means that it is relevant whether the tool is useful in the customer’s environment not whether it is useful in a fictitious software process under fictitious conditions. Figure 1 illustrates the value as perceived by the customer.
The x-axis represents different quantities of a product characteristic (in our example: a CASE tool’s functionality). The y-axis represents the normalized value a potential customer assigns to different quantities of the product characteristic. Curve A shows the dependency of the value as perceived by the customer from the amount of functionality offered by a classical PC-CASE tool, supporting the early phases of the software life cycle (upper CASE tool). Curve B shows the dependency for a Client-Server CASE tool offering various generation tools and supporting a wide range of phase independent activities. The value 100 is reached for both products at a rather different functionality. A lack of functionality in respect to this required functionality in both cases means little value for the customer. An increment of functionality in respect to the required functionality does not increase the perceived value proportionally. Therefore the different levels of functionality characterize different kinds of CASE tools not different qualities in the sense of „goodness“. The value of a CASE tool for a certain customer depends on his specific requirements. The CASE tool evaluation study of the University of Cologne therefore carefully avoided any hint to a ranking of the tools. This also points out that customer-orientation is a critical success factor for a CASE tool producer.

If a customer’s requirements are not met, he will not buy the product or he will demand a substantially lower price (severe punishment of underachievement). The curves have a steep slope on the lower side of the re-
quired functionality. On the other hand overachievement does not mean a proportional increment of value for the customer: A small or mediums size enterprise would not be willing to pay a higher prize for a complex CASE tool offering a lot of functionality the small enterprise would not take advantage of. At least it is unclear whether the excess cost of overachievement will reach amortization. Misunderstood „positive“ quality at best means an increased risk. Overachieving customer requirements makes sense only in exceptional cases for example if all customer requirements are met. The goal of product planning should be to meet customer expectations as close as possible. This is what QFD supports.[3]

Product planning of a CASE tool in contrast to planning of other software products like text processing for example has some particularities, which have to be considered in the application of QFD:

- product complexity

Because of a CASE tool’s complexity many customers cannot be expected to formulate true needs and to distinguish reliably requirements of major and minor importance.

- application complexity of the product

Applying a CASE tool is a very complex task, since it means automation of the very complex and not well understood process of software development. There are many factors and unknown conditions influencing the successful application of a case tool. As a consequence the customer requirements are sometimes conjecture or mere guesswork and software engineering specialist claim to know better.

- external factors

Technology trends (for example object-orientation, client-server architectures) and management trends (for example total quality management) usually cause novel requirements upon CASE tools. Unfortunately these trends have effects on the product that are neither foreseeable nor controlable by neither the customers nor providers.

- quality versus innovation

Another typical obstacle to customer-orientation of CASE tool producers is the different evaluation of quality requirements and innovation by de-
developers and clients. While clients and near-to-client-departments of the producer give high priority to quality, developers tend to give higher priority to innovations. Software engineers usually tend to see themselves as inventors of fundamentally novel ideas and not primarily as developers of useful products. Therefore we frequently find a hidden competition between quality and innovation, in which the looser is quality. Often quality lacks a promoter in the development department.

Therefore QFD is an instrument with a twofold effect:

- it supports product planning on the basis of the customer’s voice by a stepwise analysis and deployment of customer requirements and
- it supports an improvement of communication between all people involved in a product development.

However, there are two other effects it will not exhibit: the control and prediction of customer requirements and trends.

3. QFD as an Instrument for Improving Customer-Orientation

Much of the success of Japanese companies is commonly attributed to their employment of simultaneous engineering. Simultaneous engineering uses a synchronous definition of product-design and production-processes based on customer’s requests and interdisciplinary team work in the company. QFD is the backbone of simultaneous product-planning. [4] From pronounced, provided or latent customer wants and needs relating to the projected product or improvement the product’s features are successively developed. During the first phase „quality-planning“ customer-requirements are mapped to technical characteristics of the product. In the following multistage phase „part-deployment“ the characteristics of the product-parts (modules for example) are derived from the product’s features. During the „process-planning“ phase the part characteristics and eventually specific process requirements are used as important criteria for the definition of key process operations (object or time of testing for example).

Any important technical and economic data as well as the fundamental data for defining the market strategy are fixed by an interdepartmental team of experts in marketing, product-planning, product-development and quality
management, etc. Every step of planning is recorded in a QFD planning-matrix, which consequently represents a kind of project summary.

Applying QFD in the development of new and improved software products could yield several advantages:

- it gives development a clear focus on customer requirements
- it supplies management with an early and clear survey of the critical issues and conflicting goals
- it provides a rational and transparent basis for development decisions
- it supports developers with clear guidelines for software process organization

Therefore it is of great interest to adapt QFD for use in software development either. There have been some attempts to do so [5], but there is not yet a stable methodology.

4. Potential and limits of QFD in the development of CASE tools

We shall discuss the potential of QFD for planning software development employing a short example of the planning of a CASE tool.

During the first phase of QFD, quality planning, quality requirements are determined from the voice of the customer. Traditionally two types of requirements can be distinguished: functional and non functional requirements, where usually the non functional requirements attract only little attention. Among the non functional requirements particularly quality requirements are virtually neglected. This is in sharp contrast to the fact that in empirical studies CASE tool users give priority to quality requirements. [6] QFD can be used for deployment of functional as well as non functional requirements. In surveys unsatisfied functional requirements are of minor importance in the explanation of customers’ dissatisfaction. Therefore we give priority to the improvement of the process of deploying quality requirements. Customer requirements need to be sufficiently precise and detailed in order to form a solid base for decisions about technical characteristics of the product. As has been pointed out in chapter 2, especially for users of CASE tools demanding precise and detailed quality requirements means
expecting too much. This can also be demonstrated in the process of selecting a CASE tool, where a product assessment catalogue is needed. In order to save the effort of systematizing their requirements many companies find it very helpful to base their own assessment on a standard CASE tool assessment catalogue [7] containing potential requirements in a systematic and precise form. In a first step customer requirements have to be transformed into a system of precise and detailed requirements. To prevent bias this should be done with the customer, who also has to attach his weights to the final requirements. In planning the first release of a product the dominating direction of development of the customer requirements will be from abstract to specific. In planning further releases the customer will express very specific requirements from his past experience of the product that need to be organized, aggregated and abstracted.

In the planning of a new release of a simple information system we used moderated sessions with customers. The sessions were based on some simple rules. The „producers“ were allowed to ask customers for information, explanation or detail. They were not allowed to comment on requirements, wishes or complaints of customers. The customers were encouraged to detail, interpret, explain and discuss their requirements. They were further encouraged to describe their use of the product and to comment on the product, its strengths and deficiencies. The results of the group sessions were consistent with an independent survey, that has been conducted before. But the moderated sessions yield additional surprising requirements. Because of the great amount of requirements priorization is difficult. It seems to be helpful to use a pairwise comparison. Of course this judgement needs to be considered carefully. The priorization of the requirements can only be used as basis for development decisions, if the selection of customers participating in the sessions was sufficiently representative.

In detailing requirements one can proceed similarly as in a traditional development of a quality model. Quality requirements like for example user friendliness are separated into aspects (secondary requirements) like ease of use, short period of training, individual adaptability (figure 2).
Figure 2: Typical customer requirements for a CASE tool

If necessary the secondary requirements can be further broken down into tertiary requirements and so on. For example ease of use can be separated into ease of determination of the appropriate work context, ease of selection of the needed function, ease of input of the necessary data for the selected function, etc. Here it is important, that the detailing is solution independent and strictly from a customer’s point of view. The degree of detail (grain size) has to be determined pragmatically. It is adjusted appropriately, if the system of requirements 1. can be handled appropriately within the typical QFD diagrams, 2. allows for a detailed comparison with the quality of competing products and 3. maps easily to the components of the product.

The customer’s satisfaction with the product respectively the comparative evaluation of competing products on the right hand side of the diagram is another important information of the first QFD diagram.

During the next step the developer has to deduce product quality elements from the customer’s requirements. In this step requirements are mapped onto the product and gain precision. Quality elements should have a precise and ideally operational meaning. Usually there is no simple relation between customer requirements and quality elements. Therefore it is appropriate to describe the relation in form of a typical QFD diagram. In case there is a great number of quality elements it is advised to aggregate them into higher
abstractions. In the example the requirements „easy to use“ are mapped onto
the operational quality elements number of user errors per day, number of
steps for a typical task, time to learn for professional etc.

The quality element’s target values in figure 3 are average values and are at
the same time development goals. The target values as well as the expected
complexity (from 1 = easy to 3 = difficult) and the importance of the quality
elements at the bottom of the diagram are subjective estimations. In the roof
of the „House of Quality“ positive and negative correlations are given be-
tween the various quality elements (figure 3).

Figure 3: QFD quality planning of a CASE tool [8]

This first QFD diagram is related to the CASE product. The further dia-
agrams are related to the product’s components and improvement measures.
The focus of further analysis in our example may be the beginner as a spe-
cific type of user or the average number of steps needed for a typical task as
a particular quality element.

In component planning the same diagram technique is employed to deter-
mine which product component (figure 4) contributes to the most important
quality elements identified in the first step. For software applications a
stronger modularization is necessary as for typical industry products. Furthermore software products need not be hierarchically structured into components the same way typical industry products are.

![Figure 4: Components of a CASE tool](image)

As a component of a CASE tool we can for example focus on the entity relationship diagram editor and its functions create, delete, modify diagram or we can consider the menu system and options to reach the quality elements’ target values.

In the following steps specific design decisions are prepared concerning for example the menu structure or the amount of mouse operations.

5. **Avoidable Problems in the Introduction of QFD**

QFD has been developed in Japan by Akao. The first application was car production at Toyota [9] and manufacturing is still the most important domain of application. However, meanwhile QFD is also increasingly applied in the Japanese software industry.[10]

Capers Jones claims that US leading edge companies in information technology (Motorola, HP, AT&T) employ measurement and techniques like formal inspections and QFD as critical factors of their success. [11]
However, many attempts to introduce QFD have failed. [12] In most cases this can be attributed to the lack of necessary prerequisites particularly a change of the corporate culture towards improved customer-orientation (TQM) and the availability of useful data (about customer needs, customer satisfaction, quantified quality characteristics, target values for quality characteristics, etc.). Important reasons of failure for European users of QFD turned out to be deficiencies in cross functional cooperation. Because of the complexity of QFD, which is often underestimated, introducing QFD should start with an appropriately small pilot project. From practical experience we see that sessions for developing the matrices take two to three days for five to seven people. Collection and preparation of the necessary input information can mean a multiple of this effort depending on the availability of data in the enterprise.

6. Conclusions for QFD Application in Software Development

The above proves the applicability of QFD for the planning of software products in principle. It also demonstrates peculiarities of the application of QFD to software. The software we are interested in is used to support some business process and in case of CASE software there are very different ways to use the tool. Therefore it is important to start the analysis of customer requirements with the customer’s requirements on the business process. [13] There is also evidence for the usefulness of a QFD in the development of individual software and the customization of standard software. In such development QFD introduction may even profit from the usually closer contact and cooperation with the customer. For development of individual software products employment of QFD will also have the software process as an object. Typical customer requirements in this case are reliable estimation of effort, early availability, to have a say in development or ISO 9000 conformity of the development process.

However, in sharp contrast to the situation in the manufacturing industry quantification of process parameters, quality elements and product characteristics lack far behind in software industry. Therefore an essential contribution to the success of QFD can be expected from research and practical experience of measurement in software development.
Literature


