A framework for Quality Function Deployment

W. Pietsch

ExperTeam GmbH Consulting+Training+Software+Systems,
D-50829 Cologne, Germany

Abstract

Organizations considering the application of QfD are faced with several questions: What are the capabilities of QfD, and how does it compare to QM techniques in practice? Several introductions and application reports on QfD have been published recently; however, there is a lack of conceptual work. Therefore it is difficult for a potential user of QfD to evaluate its potentials and limitations. In order to fill this evaluation gap, this article will analyze the essence of QfD and present a framework for its application to software development.

1 Introduction

Quality Management aims at the best value of a product for the customer with respect to commercial success. A number of instruments for Quality Management (QM) have been developed so far: Value Analysis, FMEA, Statistical Process Control (SPC), Quality Circle, and Quality Function Deployment (QfD). QfD has attracted the most attention in practice recently.

Quality Function Deployment (QfD) was developed in Japan and emerged as a vital component of Quality Management (QM) for industrial manufacturing. It was transferred to the USA and applied successfully there (for instance at General Electric) as well as in Europe (for instance by Ford Europe). Organizations considering whether to implement QfD must answer a number of questions: What are the capabilities of QfD? How does it compare to QM techniques in practice? Which data is required and produced, which functions does it support? Is dedicated tool-support necessary? Several introductions and
application reports to QfD have been published recently, just a few of them deal with QfD for Software Development [5,7]. There is a vast scope of different applications using mutually incompatible terminology and producing results whose relevance, as a rule, is limited to the specific context of each individual application.

2 Elements of QfD

QfD was developed to integrate the customer’s perspective on the product into the entire production process; quality performance at several levels is compared with the strongest competitor. There are several approaches to QfD employing different numbers and types of charts and procedures. Specht/Schmelzer propose the following QfD-process [6] (see also [2,3]):

1. Quality Planning: Customer needs are 'translated' into technical attributes of the product (Quality attributes).

2. Component Design: Requirements for the Components used for the manufacturing of the product are defined on the basis of the Quality attributes.

3. Process Planning: Technical attributes of the components are translated into critical product parameters and process parameters.

4. Process Design: Work plans are specified, that satisfy the product/process parameters.

At each stage, one set of criteria (Customer Needs, Quality Attributes, ...), referring mostly to some kind of requirement, is linked or associated to another set of criteria (Quality attributes, Component Requirements), which refer to some kind of solution. A table is used to document the association: each row refers to one set of values in rows, and the columns to the other set of values, the body contains the associations. Such a table is developed at each stage; figure 1 gives an example.

The associations are often refereed to as correlations; they denote the subjective assessment of the strength of the interactions between a solution-alternative pair by domain experts. A triangle represents a ‘weak’ correlation, a circle represents a ‘medium’ correlation and a dotted circle stands for ‘strong’ correlation [1,2,3,4]; the numerical values 1,3, and 9 respectively are used to quantify the correlation. Since criteria are rarely independent of each other, one or two half-matrices denoting correlation among criteria is attached to the Quality table, as shown in figure 1. The resulting chart rather resembles a
house in elevation view; for this reason, quality tables are also known as ‘House of Quality’ (HoQ) [1].

![Sample 'House of Quality']

Since the HoQ of two subsequent stages share one specific criterion, the HoQ of all stages may be combined into a System of QfD charts. The left part of Figure 2 gives an overview of such a System. The System of QfD charts can then be used to transform user needs into manufacturing processes (by way of quality attributes, ...), making it possible to relate each manufacturing process to user needs. The right half of Figure 2 shows a sample trace following the correlations through three HoQ's.

The trace feature of QfD facilitates the integration of requirements and solutions on several levels of description and for different user groups. Whereas marketing is primarily interested in user needs and recognizable quality features, manufacturing is interested in processes and mechanisms. The HoQ supports not only each view by itself but also its integration with other views. It is also remarkable in that it reflects not only those features that can be quantified or clearly specified, but also vague criteria, fuzzy criteria, which can at best be described with a linguistic expression. This preserves QfD from the pitfalls of a reductionist view on Quality. It is a valid approach, since the HoQ is in-
tended to be changed and manipulated only by domain experts; it is devised for communications between different domain experts and not for ‘end users’.

Figure 2: System of QfD charts

A second feature of QfD is Target Costing. Criteria are analyzed not only by themselves, but also with relation to costs and performance of the best competitor. QfD may be considered as a very sophisticated form of cause effect analysis, augmented with a target costing approach, which must be tailored to the specific business environment. In order to form a generic framework, which can is readily customizable, elementary data and then elementary functions using these data must be identified. To this end, an elaborated QfD approach has been analyzed thoroughly.

3 QfD Data

Figure 3 classifies King's elaborated QfD system [4], consisting of 32 charts. The charts may be broken down into the following types: correlation charts (HoQ), comparison charts, attribute lists, and other kind of quality charts, e.g. failure effect analysis. Each chart is given a short name according to its position in the system of charts, with a number for the column and a character for the row. Figure 3 depicts the resulting 'Table' of Charts.

The QfD charts referred to in Figure 3 refer to different data elements ranging from product data to cost and performance data. In transferring QfD to an application area like software development, the precise attributes themselves are secondary; before anything else, the scope of the tables must be understood. The following figure 4 gives a generic classification scheme for criteria for QfD
charts. Each of King's charts, as well as all other QfD charts the author was able to examine, may be classified and thus also constructed by a restricted set of criteria. Figure 4 presents a tentative specification of a generic criteria set for QfD applications to manufacturing.

Figure 3: Table of QfD charts

<table>
<thead>
<tr>
<th>Criteria for Manufacturing</th>
<th>Component</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Part</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mechanism</td>
</tr>
<tr>
<td>Fault</td>
<td></td>
<td>Quality Attribute</td>
</tr>
<tr>
<td>Requirement</td>
<td></td>
<td>Customer Need</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Function</td>
</tr>
<tr>
<td>Process</td>
<td></td>
<td>New Approach</td>
</tr>
<tr>
<td>Innovation</td>
<td></td>
<td>Improvement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New Technology</td>
</tr>
<tr>
<td>Supplier</td>
<td></td>
<td>Competitor</td>
</tr>
<tr>
<td>Producer</td>
<td></td>
<td>The Company Itself</td>
</tr>
</tbody>
</table>

Figure 4: Generic Criteria Set for QfD in Manufacturing
424 Software Quality Management

In order to develop an individual QfD system one has to decide, which criteria are specifically important to a particular business within its context. Then one has to analyze how these criteria are used in the product planning and manufacturing process - or better - how these could be used for effective communication with regard to customer satisfaction and costs. The result is a system of QfD charts, which may now be augmented with planning data, such as performance measures, control levels, cost data, and so on.

4 Elementary QfD Functions

In order to apply the system of charts to QM, different kinds of functions must be performed on these charts. An in-depth analysis of QfD leads to the result, that they may be characterized using the following categories:

1. Calculation: the parameters and result type are numerical. The result is computed from the parameter by means of arithmetical operations, for instance:
   
   1.1 *Scalar Multiplication* of the values in a column with the values of another column;
   
   1.2 Calculation of a *percentile* of one attribute in relation to another attribute (implies summing up the values of all attributes of the latter);
   
   1.3 *Aggregation* of correlations with other criteria, calculated by summing up the numerical value of the correlation with the attributes of another criteria;
   
   1.4 *Counting* 'Plus'- and 'Minus'-values for the attributes of a criterion to be compared to another criterion.

2. Analysis: The result of a QfD-analysis is not numeric, it is the result of an evaluation of the characteristics or the structure of the QfD-table under consideration (for instance, search for requirements) that correlate to 'many' solutions of another specified criterion.

   2.1 *Evaluation*: Analysis is performed on a complete correlation chart (HoQ), with the result generally qualitative; for instance, the most important attribute may be asked for.

   2.2 *Range analysis*: The structure but not the values of the Charts is analyzed; they are applied by the user in order to cross-check charts and to unveil inconsistencies.

It is obvious that the power of QfD lies not in its functions, but rather in its tailoring to a specific QM system. The functions described above are easily supported by a conventional spreadsheet program such as Microsoft Excel, so
that even the hierarchical structuring of criteria, another feature of QfD, can realized quite simply.

But QfD is anyway generally performed manually, for several reasons. QfD charts in practice tend to become very large in practice, and the data they contain is usually rather ‘sparse’. The human eye can use the redundancy within a large table for ‘pattern matching’ and in order to recognize global structures, e.g. clusters or missing values among others. Tools, as a rule, will support simple evaluations such as empty columns, but not ‘logical clusters’ or the generation of other such hypotheses in the course of an intelligent evaluation. Furthermore, it is difficult to ‘reconstruct’ the handling of a large paper-based chart under the restriction of a ‘window’.

However, manual handling of QfD charts is error-prone, inefficient and inadequate for data and process integration. There is therefore an emergent need for dedicated QfD tools. Nevertheless, QfD tools support primarily the technical and not the conceptual part of QM, especially in its tailoring. Following modern Design practice, a conceptual scheme for the application of QfD must first be developed. The generic scheme described above and the elementary functions may be considered as generic building blocks for the design of a QfD Architecture tailored to any domain, and especially for software development applications, as shown below.

5 Application to Software Development

The generic criteria set for manufacturing is the starting point for the transfer of QfD-concepts from manufacturing to software development. The first-level concepts like parts, requirements and faults may easily be transferred to software development. Figure 5 shows a generic criteria set for the application of QfD to software development.

A ‘part’ in Manufacturing is to be compared with a ‘system unit’, such as ‘software products’, ‘modules’, ‘data’, or ‘procedures’. The unit type chosen depends on the development method and the level of quality planning appropriate for the specific environment. The concept ‘fault’ is well-known to software development, but mostly as a control entity and not as a planning entity, except as part of the ‘Zero Defects’ approach. ‘Requirements’ are the prevalent planning entity for software development. However, these requirements are generally associated with the notion of a given, stable definition of ‘functional elements’ of a prospective system to be implemented; the original ‘needs’ of the end user are often neglected.
The (sub-)process in manufacturing corresponds roughly to a software development ‘task’. Innovations in software developments are triggered by new design approaches, new technologies or ‘perfective’ maintenance activities. A straightforward interpretation of innovation as in manufacturing is not adequate, since research, development and production have a different meaning in the two areas. The same problem arises with the translation of the concept of a ‘supplier’; there are no suppliers of ‘raw material’ within software development, but there are sub-contractors or other partners, e.g. for outsourcing. In some cases, there may not even be a ‘competitor’; monopoly has been common in traditional in-house development, but not in today’s open software market.

The transfer of concepts made clear that besides the facilitation of the tracing of product attributes QfD has been developed for an open market, a consequent orientation to the needs of the customer - a situation that is not applicable to software development yet, but will surely emerge in the near future. A cultural change within software development is the prerequisite for the success of QfD. The introduction of a new technique like QfD may support or enable, but will not foster the change process. Knowledge about QM is mandatory for QfD.

Knowledge of QM is needed in order to select the most appropriate QfD-criteria and combine them into a valid QfD system. If a company has already implemented a modern QM system without QfD, and experienced its advantages and limitations, then the introduction of QfD should not pose any more problems than that of any other technique. If not, then there is a long way to

<table>
<thead>
<tr>
<th>Criteria for Software Development</th>
<th>System Unit</th>
<th>Software Product</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Module/Data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Procedure</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Software Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>User Need</td>
</tr>
<tr>
<td></td>
<td>Functional Element</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Development Task</th>
<th>Innovation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>New Design</td>
</tr>
<tr>
<td></td>
<td>Perfective Maintenance</td>
</tr>
<tr>
<td></td>
<td>New Technology</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sub-Contractor</th>
<th>Producer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Competitor</td>
</tr>
<tr>
<td></td>
<td>Development Team</td>
</tr>
</tbody>
</table>

---

Figure 5: Generic Criteria Set for Software QfD
the introduction of QM and QfD. QfD can only be introduced in small increments.

Figure 6: A System of QfD charts for Software Development

(1) A first step might be the design of a single HoQ for functional elements and components, which may be derived from the system documentation when using integrated development tools (CASE). Then for each functional element the quality metric and the control level must be defined, and for each component the projected and the target costs determined.

(2) In a second step, functional elements can be related directly to user needs. For each specific user need performance metrics and target values must be specified.

(3) In the third step, the focus could be set on the development process itself. For this purpose the tasks must be defined and related to system units. Figure 6 depicts the resulting QfD System which is comprised of three
Software Quality Management

HoQ, associating four criteria: needs → functional elements → units → tasks.

If tasks and units are either standardized or derivable from a generic set of building blocks, then this QfD system would be capable not only of supporting an integrated quality view and facilitating concurrent development. It would also make possible economic analysis of problems and solutions before a detailed analysis has been performed: design to cost, requirements scrubbing, and finally target costing could all be achieved. Economic analysis is an integral feature of QfD for manufacturing; experience will show, whether this is also valid for software development.

References