CSIRO Australia
Division of Manufacturing Technology

The University of New South Wales

PROJECT REPORT

on

Customer Driven
Computer Assisted
Product Development System **for**
Quality Function Deployment

Sydney, February 1997

presented by:

Matthias Runte
Alte Rennbahn 35
21244 Buchholz
Germany
Runte@tu-harburg.d400.de

Preface

This report is submitted to Simmy Grewal, project manager of Concurrent Engineering, CSIRO, Division of Manufacturing Technology, Sydney Laboratory.

The presented work is part of the current Concurrent Engineering project. It is directly based on the work of Ingo Westphal from May 1996.

The work was supervised by Dr. Leonard Farmer, University of New South Wales.

Acknowledgement

I would like to thank Dr. Leonard Farmer, Prof. Dr. Hartmut Kaebernick and Dr. Simmy Grewal for enabling me to write my diploma thesis at the facilities of CSIRO Australia, Department of Applied Physics, Division of Manufacturing Technology.

I gratefully acknowledge the continuous support from Dr. Leonard Farmer for his help in word and deed.

I would also thank Ingo Westphal, who formed the base for this thesis; Curtis, who persistenly searched for every peace of literature I needed; Brenda Gale, for her advice and administrative aid; Cong, who supplied me with hardware and software; Mark, who helped me solving some mechanical problems; Narelle, who carefully read the thesis and corrected some of the countless English flaws; the entire team of CSIRO's Division of Manufacturing, who helped me wherever it could.

They all deserve my grateful thanks and made my stay in Australia an experience that I will treasure in my whole life.

Table of Contents

Preface .... *II*

Acknowledgement .... *II*

Table of Contents .... *III*

Table of Figures .... *V*

List of Abbr. .... *VII*

1 Context, Objective and Scope of this paper *1*

2 Databases for QFD Projects *3*

2.1 Databases **3**

2.2 Relational Databases **4**

2.3 Cardinality **5**

2.4 Database Architecture **6**

2.5 Entity-Relationship Diagrams **7**

3 The Voice of the Customer *10*

3.1 Who are the customers? **10**

3.2 The Kano Model **10**

3.3 Obtaining the Voice of the Customer **12**

3.4 Qualitative Customer Survey **13**

3.5 The VOCALYST process **14**

3.6 Summary of the Voice of the Customer Input Process **15**

3.7 Computer Support for the VOC Input Process **16**

3.7.1 Provided Functions in the QFD-Support System 16

3.7.2 Extraction of Voice of the Customer 16

3.7.3 The Merging-Module 18

3.7.4 The VOC Data Tools 20

4 The Affinity Process *22*

4.1 Controlling the Number of Requirements **22**

4.2 The Affinity Diagram Process **22**

4.3 Application of the Affinity Process for QFD **24**

4.4 Computer Support for the Affinity Process **25**

4.4.1 Benefits of Computer Support 25

4.4.2 Maintaining the Overview 26

4.4.3 The QFD-Manager Affinity Module 28

5 Quantitative Survey *30*

5.1 Importance Ratings **30**

5.1.1 Quantifying Customer Requirements 30

5.1.2 Ranking Systems 30

5.1.3 Critical Customer Requirements 36

5.2 Customer Competitive Assessment **37**

5.3 Computer Support for the Quantitative Survey **38**

5.3.1 Printed Questionnaire 39

5.3.2 Diskette Survey 39

5.3.3 Internet Survey 40

5.4 Data Structures for Computer Assisted Surveys **43**

6 Charts in QFD *46*

6.1 Charts and Matrices **46**

6.2 The need for a Pre-planning Chart **46**

6.3 Calculation Methods for Requirement Weights **48**

6.3.1 Multiplication of Scores 48

6.3.2 Multiplication of Ratio Scores 49

6.3.3 Addition of Scores 49

6.4 Computer Support for QFD-Charts **50**
6.5 Chart-Handling in the QFD-Manager **51**
6.5.1 The Field-Manager 51
6.5.2 The Data-Manager 52

7 The Technical Information Table ***53***
7.1 Translating Requirements into Characterisitics **53**
7.2 Improvement Directions **55**
7.3 Technical Competitive Data and Target Values **56**
7.4 Summary of Generating Technical Characteristics **58**
7.5 Computer Assistance for Technical Characteristics **59**

8 Matrixhandling ***62***
8.1 Charts and Matrices in QFD **62**
8.2 Necessity for Computer Support **63**
8.3 The Matrix Module **64**
8.4 Functions of the Matrix Module **65**
8.5 The Relationship-Manager **66**
8.6 Structure of the Relationship Table **67**
8.7 Field and Matrix Calculation **68**
8.7.1 Field Formulas 68
8.7.2 Formula Syntax 69
8.7.3 Formula Evaluation Algorythm 71
8.8 Graphical Display and Matrix Print-Outs **75**
8.8.1 The Need for Printing out Matrices 75
8.8.2 Sections and Elements of QFD Matrices 76
8.8.3 Matrix, Screen and Printer Coordinate Systems 77
8.8.4 Related Problems and Solutions 80
8.9 Matrix Deployment **81**
8.10 Matrix Module Summary **82**

9 Concepts ***83***
9.1 Concept Generation **83**
9.2 Computer Support for Concept Generation **86**
9.3 Concept Selection **86**
9.4 Computer Support for Concepts Selection **88**

10 System Overview ***92***
10.1 Conceptual Database Design **92**
10.2 Logical Database Design **94**
10.3 Future Developments **96**

11 Summary ***98***

12 References ***99***

Table of Figures

Figure 2-1 - Example for Tables in a Relational Database 4
Figure 2-2 - Aggregation of two tables 5
Figure 2-3 - Database Field Formats 5
Figure 2-4 - The four basic Types of Relations 6
Figure 2-5 - Database Architecture Levels 6
Figure 2-6 - Example of an Entity-Relationship diagram 8
Figure 3-1 - the Kano model 11
Figure 3-2 - The Three Categories of Requirements 12
Figure 3-3 - Data structure of Customer Statements 18
Figure 3-4 - ER Statements, Masterfile, Workfile 19
Figure 4-1 - Affinity diagram 23
Figure 4-2 - Structured Requirements for a Coffee Cup 24
Figure 4-3 - Memory Jogger Plus - Affinity Sample 26
Figure 4-4 - QFD-Manager - Grouping Module 27
Figure 4-5 - Data structure for Affinity Module 28
Figure 5-1 - Self-explicated Ranking 31
Figure 5-2 - Constant-Sum Ranking 32
Figure 5-3 - Anchored Ranking 33
Figure 5-4 AHP Ranking Scale 34
Figure 5-5 - AHP Evaluation Matrix 34
Figure 5-6 - Normalised AHP Evaluation Matrix 34
Figure 5-7 - Cascaded AHP 35
Figure 5-8 - Customer Competitive Assessment 38
Figure 5-9 - Criteria for Survey Media 39
Figure 5-10 - Internet Hosts 1989-1996 41
Figure 5-11 - Internet Hosts - Overall Trend 41
Figure 5-12 - Dataflow in the QFD-Manager 42
Figure 5-13 - Internet Survey displayed by a Web-Broweser 43
Figure 5-14 - Data Structure of Survey Definition Table 44
Figure 5-15 - Data Structure of the Survey Data Table 45
Figure 6-1 - Customer Table and Technical Table 46
Figure 6-2 - Pre-planning chart, calculated by Multiplication of Scores Method 48
Figure 6-3 - Pre-planning chart, calculated by Multiplication of Ratios Method 49
Figure 6-4 - Pre-planning chart, calculated by Addition of Scores Method 50
Figure 6-5 - The Field Manager 51
Figure 6-6 - The Data-Manager 52
Figure 7-1 - The Main Tables of the House of Quality 53
Figure 7-2 - Translating Requirements into Characteristics 55
Figure 7-3 - Symbols for Improvement Directions of Characteristics 56
Figure 7-4 - Results from technical competitive assessment 57
Figure 7-5 - Development of Technical Table 58
Figure 7-6 - Structure of Characteristics Table 59
Figure 7-7 - The Characteristics Module 60
Figure 8-1 - Systematic of Matrix Calculation 62
Figure 8-2 - The Matrix Module 64

Figure 8-3 - Matrix Definition Table 65
Figure 8-4 - The Relationship-Manager 67
Figure 8-5 - Relationship Crosstable 68
Figure 8-6 - Formula Elements 69
Figure 8-7 - Function Set of QFD-Manager 69
Figure 8-8 - House of Quality Calculation Example 71
Figure 8-9 - Flowchart of Evaluate Field Value 72
Figure 8-10 - Flowchart of Evaluate Function 73
Figure 8-11 - Flowchart of Table Calculation 74
Figure 8-12 - Flowchart of Matrix Calculation 75
Figure 8-13 - Sections of a QFD matrix 76
Figure 8-14 - Matrix Elements 76
Figure 8-15 - Transformation from Matrix to Screen Coordinate System 77
Figure 8-16 - Transformation from Matrix to Printer Coordinate System 79
Figure 8-17 - QFD-Manager Print Preview 80
Figure 8-18 - Matrix Deployment 82
Figure 9-1 - Car Horn Concepts 85
Figure 9-2 - Concept selection matrix 87
Figure 9-3 - Concept Functions 89
Figure 9-4 - Concept Selection Matrices in the QFD-Manager 90
Figure 9-5- Concept Selection Process using QFD-Manager's Deployment Functionality 91
Figure 10-1 - QFD-Manager Entity-Relationship Diagram 93
Figure 10-2 - Container Table 94
Figure 10-3 - Requirements Table 95
Figure 10-4 - Characteristics Table 95
Figure 10-5 - Masterfile Table 95
Figure 10-6 - Matrix Definition Table 95
Figure 10-7 - Relationship Table 95
Figure 10-8 - Survey Definition Table 96
Figure 10-9 - Survey Data Table 96

List of Abbr.

Abbr.	Abbreviation
ASI	American Supplier Institute
Card	Cardinality
DBMS	Database Management System
DDL	Data Definition Language
DML	Data Manupulating Language
DQL	Data Query Language
eg	example given
FA	Function Analysis
FAST	Function Analysis System Technique
FMEA	Failure Mode and Effect Analysis
FTA	Fault Tree Analysis
GOAL/QPC	Goal/ Quality, Productivity, Competitiveness
ie	in example
max	maximum
min	minimum
QFD	Quality Function Deployment
VB	Victoria Bitter
VOC	Voice of the Customer
VOCALYST	Voice of the Customer Analysis System Technique
VOCT	Voice of the Customer Table

1 Context, Objective and Scope of this paper

This Master's thesis is written in the context of the Lifecycle Management Project of CSIRO (Commonwealth Scientific and Industrial Research Organisation) Australia. This paper is directly based on the thesis of Ingo Westphal,[1] part of a series of Quality Function Deployment (QFD) related theses at CSIRO's Lindfield site, Division of Manufacturing Technology, in co-operation with the University of New South Wales (UNSW). His work shall be reviewed, refined and extended where necessary. A QFD software shall be developed that may act as a base for high-end commercial QFD Software.

Westphal begins his work with a comprehensive overview of QFD, its elements and different approaches. He then focuses on special steps of QFD studies, which he uses to set up a *"New Road Map for QFD Applications"*. Having accomplished this, he discusses possibilities for Computer Support for the different QFD Steps, which finally end in the two major parts of his developed software prototype *"QFD-Support-System"*. The two modules are called *Voice-of-the-Customer* Module and *Matrix-Handling* Module.

Additionally to Westphals work, this paper is also based on at least four preceding thesis which deal with QFD. This thesis is meant to be read in the context of these papers. For that reason, the introduction of Total Quality Management (TQM) and QFD in general, which occurs in all these papers, will be omitted to avoid repetition.

Westphal states that his software can not claim to be a professional software. He writes about his prototype: "Still, the test program is just a first attempt to create a QFD support system. Many functions are missing. Some can be replaced provisional by other programs which can access and use the data. [..] If a comprehensive QFD support system [..] could be created successfully, it would probably increase the overall number of QFD applications and, more important the number of successful QFD applications."[2]

The aim of this thesis is to present computer software which can act as the base for a high-end computer system that is competitive on the QFD software market. The work title of the new software is ***"QFD-Manager"***.

In the thesis, the system will be developed step by step. With each step, a different part of QFD projects is discussed. Possible computer support is evaluated and the implemented modules of the new system are presented. This varies from Westphals approach, which describes respectively all of the different QFD steps, special elements, the New Road-Map, possible computer support and finally the implementation of his software. The author of this thesis believes that his approach is "easier to swallow" for readers not particularly familiar with the QFD methodology. This is because single parts or tasks of QFD are discussed directly together with offered computer support.

In chapter 2 an introduction to relational databases is given. This is necessary because almost all data structures of the system are based on the relational model. Chapter 3, 4, and 5 describe the gathering of customer voices and their development into structured, evaluated and quantified customer requirements. Chapter 6 gives an overview of charts that are used in QFD as parts of matrices, presenting the example of the Pre-planning Chart. Chapter 7 describes a computer's capability to support the development of technical design characteristics.

Chapter 8 is one of the core chapters of this paper. It describes the generic methodology of QFD matrices and presents the *Matrix Module* of the QFD-Manager. In chapter 9, the concept

[1] Westphal 1996

[2] Westphal 1996, p.141

selection process as one of the most important parts of a QFD project is discussed. It will be demonstrated how to conduct the process with the QFD-Manager.

After all mosaic parts of the QFD-Manager have been described in detail, an overview is given in chapter 10. The conceptual and logical database design will be presented and the capability and constraints of the system will be discussed.

2 Databases for QFD Projects

In a QFD project a large amount of data is created. The data comprises customer requirements, design characteristics, matrix definitions, and many other items. The main purpose of a QFD computer support system is to handle the project data. Since the whole system will be focused on that data, it is necessary to focus on the best way to store this data in computer systems.

2.1 Databases

Generally there are two effective methods of data storage. Firstly, the data can be stored in a file that can only be accessed by the particular program that generates the data. Secondly, it can be stored in a database.

The original reason for the introduction of databases in 1960 was the limited memory of computers' secondary storage. Limited memory is no longer a problem, but the objectives for the use of databases have been extended by far, including[3]:

- Separation of data description and data manipulation
- Logical data independence
- Physical data independence
- Efficient processing of database operations
- Easy data administration and control
- Minimal redundancy and minimal storage space
- Data integrity
- Data sharing
- Data security

The management of data within databases consists of three tasks:

1. *Defining* the general data structure of the database using a Data Definition Language (DDL).
2. *Changing and updating* the data using a Data Manipulation Language (DML).
3. *Retrieving* subsets of data using a Data Query Language (DQL).

A system that handles all the three tasks is called a *Database Management System* (DBMS).

The three basic types of databases are the hierarchical model, the network model, and the relational model. Because of several drawbacks of the hierarchial and network model, the widest-spread of the three types is the relational model[4]. For the QFD support system to be developed the relational model will be used. It is beyond the scope of this paper to describe the other two types of models[5].

---

[3] Gardarin 1989, p.66

[4] Parsaye 1989, p. 38

[5] For a detailed description of the hierarchical and network model refer to Parsaye 1989, p. 15ff

2.2 Relational Databases

The relational model was introduced by E. F. Codd[6] in 1970, although a design of a set-oriented model had already been proposed by Childs in 1968. The first relational database system was implemented by IBM in 1976. Since the introduction of dBASE as the first relational database on a microcomputer, the relational model began to appear in the most commercial database management systems. [7]

A relational database is a database consisting of a set of tables. A table contains fields (columns) and records (rows). The fields contain the structure of the table, where as the records hold the actual information.

In the following example the table "Customers" consists of the fields "Customer No", "Name", "Street", "City" and "Phone". The table contains three records.

Table: Customers

Customer No	Name	Street	City	Phone
1	Müller, Gerd	2 Milray St	Lindfield	723 4592 234
2	Meier, Sepp	99 Lane Cove Rd	Aston	835 2385 235
3	Derwall, Jupp	77 Hughton Pd	Sydney	735 3472 237

Table: Orders

Order No	Order Date	Customer No	Delivery Date
1	19/02/96	1	30/03/96
2	20/02/96	3	30/03/96
3	21/02/96	3	30/03/96
4	22/02/96	2	31/03/96

Figure 2-1 - Example for Tables in a Relational Database

The tables of a relational database are linked by *relations*. Relations are established by certain fields in the tables. In the above example, the Field "Customer No" in the "Orders" table forms a relation to the field called "Customer No" in the "Customers" table. Relations can comprise two or more tables.

Relations are used to build up aggregations of two or more tables. The relation in the example of Figure 2-1 constructs an aggregation of the tables "Customers" and "Orders" which is depicted in Figure 2-2.

---

6 Codd 1970, in: Gardarin 1989, p. 91

7 Parsaye 1989, p. 43

Order No	Order Date	Customer No	Name	Street	City	Phone	Delivery Date
1	19/02/96	1	Müller, Gerd	2 Milray St	Lindfield	723 4592 234	30/03/96
2	20/02/96	3	Derwall, Jupp	77 Hughton Pd	Sydney	735 3472 237	30/03/96
3	21/02/96	3	Derwall, Jupp	77 Hughton Pd	Sydney	735 3472 237	30/03/96
4	22/02/96	2	Meier, Sepp	99 Lane Cove Rd	Aston	835 2385 235	31/03/96

Figure 2-2 - Aggregation of two tables

Aggregations are used to retrieve data. For example, if the phone number of the customer who ordered a product on the 19/02/96 is required (eg because the delivery date cannot be confirmed), then the aggregated table in Figure 2-2 can provide the requested information. Aggregations are constructed using *Database Queries*.

The data contained in the fields of a table can have different formats, which can vary on different operating systems. The basic field types which are commonly accepted on any platform comprise the formats displayed in Figure 2-3.

Format	Content
text	alpha numerical data, eg "Hello World!"
integer	integer value, 2 byte. Ranges from -32,768 to 32,767
long integer	integer value, 4 byte. Ranges from -2,000,000,000 to 2,000,000,000
single	numerical value, 4 byte (eg. 3.14159)
double	numerical value, 8 byte
date/time	eg. 10/03/72, 12:30

Figure 2-3 - Database Field Formats

2.3 Cardinality

Different types of relations are distinguished by their *cardinality*. The cardinality consists of two values, the *minimal cardinality* and the *maximal cardinality*.

Minimal Cardinality. The minimal cardinality of a table according to an aggregation is the minimum number of records in which an element of the table can participate.

Maximal Cardinality. The maximal cardinality of a table according to an aggregation is the maximum number of records in which an element of the table can participate.

Although any integer value is valid for cardinality, the usual values of minimal cardinality are 0 or 1, for the maximal cardinality the values are either 1 or n.

Cardinalities play a major role in relational databases. The four basic types of relations (mappings) between two general sets of data (classes) are depicted in Figure 2-4 and help to illustrate the meaning of cardinality.

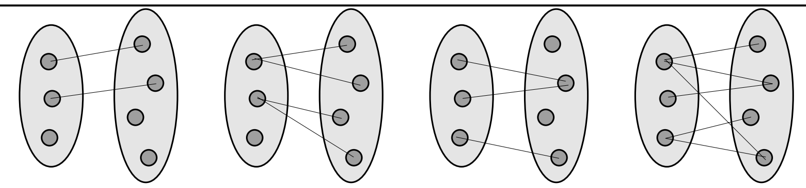


Figure 2-4 - The four basic Types of Relations[8]

One-to-One. For each element in the two sets of data (classes) exists at most one corresponding element in the other class. For instance, in Christian societies a man can only be married to one woman (at the same time), and a woman can only be married to one man. So the maximal cardinality of both classes are 1. As there may be men or women not being married, the minimal cardinality is 0. In Figure 2-1, according to the relation between the"Customer" and "Orders" tables would have a maximum and mininum cardinality of 1. This is because every order has to be assigned to exactly one customer.

One-to-many. According to Figure 2-4(b), every element of the left class can have more than one corresponding element in the right class, while every element of the right side can only have one correspondent on the left side. For instance, a man can be the father of one or more children, but a child can only have one father. The values of the minimal cardinality for the father is 0, the maximal cardinality is *n*. The value *n* means *unlimited* and is only restricted by issues of the real world, which are in the example biological reasons.

Many-to-one. The opposite case of one-to-many.

Many-to-many. The elements of both classes can have more than one corresponding element in the other class. For instance, if an aeroplane can carry 250 persons, the maximum cardinality is 250. Nevertheless, the aeroplane must be operated by at least 3 persons (pilot, co-pilot, stewardess to serve both pilots), so the minimum cardinality is 3. On the other hand, persons choose to fly with any available aircraft or they choose not to fly at all (max-card=n, min-card=0).

2.4 Database Architecture

Architecture and structure are critical issues in building databases and database management systems. There are three levels of architecture that need to be considered: The *Conceptual Level*, the *Logical Level*, and the *Physical Level*.

Level	Conceptual Database	Logical Database	Physical Database
Elements	Entities	Records	Data Definitions
	Attributes	Fields	Access Methods
	Relations	Tables	Storage Devices

Figure 2-5 - Database Architecture Levels

[8] Westphal 1996, p. 107

The conceptual view of a database is an abstract representation of the physical data and the way in which it is stored. The conceptual view is defined through a conceptual schema. In general, the conceptual schema only describes the information structures of the database, avoiding issues concerned with specific storage structures and retrieval strategies. This allows the same conceptual model of the database to apply to a variety of different physical implementations.

The development of the logical and physical database is preceded by the definition and implementation of the **conceptual database**. A conceptual database is an abstraction of the *real world* that focuses on elements of information that are relevant to the user of the database[9].

The conceptual database of relational databases is represented by the *Entity-Relationship diagram*, which will be described in section 2.5.

After the conceptual database is defined, the **logical database** is constructed. The description of the logical database consists of the definition of the logical structure of a database, written as logical schemas within a database definition language (DDL). The purpose behind defining database schemas is to specify those properties of the database that are permanently true regardless of the particular situation that happens to apply at any given point in time.

The outcome of the logical database of a relational database is a set consisting of tables which comprise fields and records. The tables can be connected logically by means of relations. The logical database is the actual description of the database structure that is created in present database software packages, for instance MS-Access.

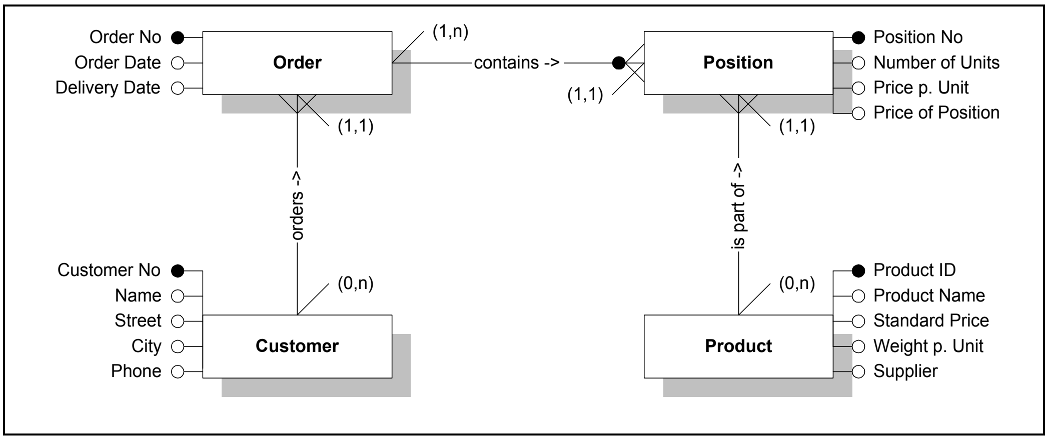
Finally the **physical database** represents the physical storage and organisation of the data. In modern databases, the database management system handles these tasks automatically in co-operation with the Operating System (OS).

2.5 Entity-Relationship Diagrams

The conceptual database is represented by the *Entity-Relationship diagram.* The elements of the Entity-Relationship diagram are *entities*, *attributes* and *relations.*

Figure 2-6 depicts an Entity-Relationship diagram with four entities:

[9] Parsaye 1989, p.49



Kommentar: ER_Example.vsd

Figure 2-6 - Example of an Entity-Relationship diagram

Entities represent classes of real-world objects. For instance, "Customers" and "Orders" are examples of entities for a database. Entities are graphically represented by rectangles (see Figure 2-6). On the logical level, entities are usually represented as database tables.

Relationships represent aggregations of two or more entities. Relationships are represented by means of diamonds. For instance, the field "Customer No" in the table "Orders" is a relationship which relates this table to the table "Customers" above. Relationships provide the functionality of aggregations on the logical level.

The two cardinality values of tables according to relations are displayed in brackets. For example (1,n) means that the table has a minimal cardinality of 1 and a maximal cardinality of n. A maximal cardinality of n is illustrated as a branch on the other side of the relationship, indicating that the entity can participate with more than one of its records in the relation.

Attributes represent elementary properties of entities or relationships. Attributes of the table "Orders" in Figure 2-1 are "Order No", "Order Date" and "Delivery Date". They are depicted in Entity-Relationship diagrams as Circles added with the name of attribute, drawed close to the table or relation respectively.

A black circle defines an attribute as an *identifier*. Identifiers are used to retrieve data from an instance. In the example, the records of the entity *Order* are identified by the attribute *Order No*. An identifier has always a unique value for each record. An attribute that acts as an identifier is called *internal identifier*.

Not only attributes, but also relations can act as identifiers. These are called *external identifiers*. In this thesis, external identifiers are displayed as black circles on the respective relation. Generally, external identifiers have a cardinality of (1,1).

Further, internal and external identifiers can be combined. For example, the identifier of the entity *Position* is its attribute *Position No* and its relation to the entity *Order*. Since a certain value of *Position No* could probably occur more than once within the entity *Position No*, only unique values for *Position No* are allowed within the same *Order*. Following this principle, the combined identifier of *Position* is unique as well. A combination of internal and external identifiers is called *mixed identifier*.[10]

Additional elements of Entity-Relationship diagrams are *composite attributes, generalisation hierarchies, and subsets.* They cannot be discussed here[11].

An Entity-Relationship diagram for the QFD-Manager will be given in chapter 10.1, after the required data structures have been derived from the Modules of the QFD-Manager.

[10] Batini 1992, p.40

[11] for a detailed discussion of ER-diagrams refer to Batini 1992, pp. 30-52

3 The Voice of the Customer

QFD is a customer-driven product development process. A main target of applying QFD is to align the product development process towards customer satisfaction. The customer requirements are pulled through all development stages.

In this paper, the quality of a product is measured by its potential to satisfy customer needs. Therefore the first key to the development of high quality products is to understand the customer needs thoroughly.

Most of the QFD systems available on the market offer a large variety and high performance to display QFD matrices. The difficult task in a QFD project is not "matrix drawing", but the generation of the matrix data. For example, this data comprises customer requirements, design characteristics, functions, concepts, relationships, and co-relationships.

In the following chapters, a computer supported method of gathering customer requirements will be discussed.

3.1 Who are the customers?

The QFD methodology is based on the philosophy that products and services should be designed according to customer requirements. Therefore, the customer is the most important part of the process. One product can have different types of customers, each with unique needs and wants.

Usually there are two basic groups of customers. The internal customers are those within in the organisation who are next in line to receive the product[12]. They may be product line people who assemble the product or service representatives who help customers install or use the product.

External customers are outside the organisation. This class comprise the final consumers of the product or distributors/retailers. Any group using or affected by the product can be defined as a customer group. For example, the customers of a petrol pump could be the filling station owners, the petrol companies, and the car drivers.

The proper definition of the customer segments is a crucial task in a QFD project.

3.2 The Kano Model

Not only customers can be divided into different groups, customers requirements can also be divided into categories.

Noritaki Kano recognized that the degree of customer satisfaction caused by the fulfillment or unfulfillment of certain customer requirements is different.[13] To be able to plan customer satisfaction, a functional relation of the fulfillment of customer requirements and the resulting customer satisfaction is required.

The Kano model is the most popular model to describe this relationship (Figure 3-1).

---

[12] Guinta 1993, p. 31

[13] Kano 1984

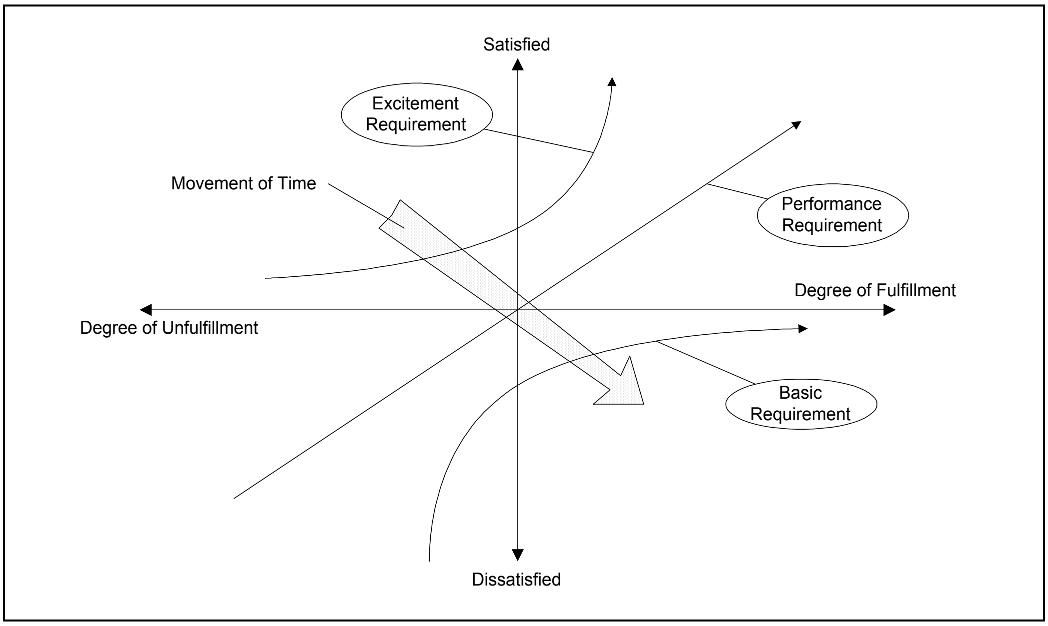


Figure 3-1 - the Kano model

In the Kano model, customer requirements are divided into three categories:

1. Basic requirements
2. Performance requirements
3. Excitement requirements

Basic Requirements. The basic qualities of the product the company must offer to be competitive are called basic requirements. These are characteristics customers assume as part of the product of service. Customers rarely ask about them as they are standard features.

For example, when you buy a car, you assume it meets all safety requirements necessary to operate the car safely. You expect also that the engine will always start without a problem.

Customers expect a company's products and services to meet minimum requirements. When minimum requirements are not met, customers take their business to competitors who can fulfill basic quality expectations[14]. Such are also called *dissatisfiers*. The fulfillment of basic requirements alone can't lead to a high degree of customer satisfaction, though.

Performance Requirements. Specific features customers desire in a product or service are called performance requirements. They are items a company is willing to provide to satisfy a customer. Performance requirements are usually communicated verbally or in writing. Purchase orders, contracts, and requests for proposal are ways in which performance requirements are communicated by customers[15]. For instance, when someone looks for a new car, he may specify the power of the engine, the maximum fuel comsumption, or the minimum volume of the boot.

---

[14] Guinta 1993, p. 33

[15] Guinta 1993, p. 33

Customer requirements are the key to customer satisfaction in determining whether the customer will purchase the product.

Excitement Requirements. Unexpected features of a product or service are called excitement requirements. These features make the product unique and distinguish it from the competition. The features may be easy or cheap to provide, as the example below shows. They may be derived from experiments, new technologies, or simply "guesses" what the customers want[16]. They often begin as unique features that later become industrial standards.

An example of a successful exciter is the cupholder Japanese manufacturers began installing in their cars in the early 1980s. A few years later this feature became standard in cars of many competitive companies. Another example is a pre-installed operating system at no extra charge on personal computers. While computers were sold without any operating system in the 1980s, retailors began to install a free operating system as a special feature. This is expected nowadays by the customer as a basic requirements.

The fulfillment of at least the basic requirements and some of the performance requirements is required to gain customer satisfaction. When some of the basic requirements are not fulfilled, the customer will not be satisfied, whether performance and excitement requirements are met or not.

In Figure 3-2, the three categories of requirments are summarised.

Category	Basic	Performance	Excitement
expected	yes	no	no
communicated	unspoken	spoken	unspoken / unknown
effect of fulfillment	none	satisfied	delighted
effect of unfillment	pissed off	dissatisfied	none
source of requirement	function analysis	customer survey	eg brainstorming

Figure 3-2 - The Three Categories of Requirements

3.3 Obtaining the Voice of the Customer

Because basic requirements are expected and therefore usually unspoken issues, it is difficult to gather the basic requirements of a product in a customer survey. They can be derived by the performing a function analysis. In value-engineering terms, the basic issues are the product's functions[17]. For example, a cup for take away coffee has the three basic functions: to hold liquid, fit hand and restrict heat transfer.

Performance requirements are communicated by the customer. In a QFD project they are usually gathered in a customer survey. The process of obtaining these requirements will be decribed in this and the following chapters.

[16] Guinta 1993, p. 35

[17] Day 1993, p. 37

It is very difficult to generate a set of excitement requirements for a product. The data is usually not obtainable from the customer because most of the exciting features are not even known to the customer. Other techniques like brainstorming sessions have to be employed. If somebody could offer a proper cook-book style method to gather all excitement requirements for products, he wouldn't waste his time on writing a thesis.

3.4 Qualitative Customer Survey

Traditional market research procedures and brainstorming by internal teams fail to provide information of the level of detail needed in QFD-based design efforts. For obtaining the true customer requirements, the real underlying need, it is essential to look behind the customers' actual words.

Techniques for identifying customer wants and needs include surveys, clinics, focus groups, individual interviews, listening (eavesdropping) at dealerships and trade shows, consumer complaints, etc.[18]

In most QFD projects the initial list of customer attributes is developed internally by some form of brainstorming. Affinity diagrams are then used to consolidate and organise the individual attributes. This process has a number of critical shortcomings. What emerges is not the Voice of the Customer, but rather the Voice of the Company[19].

Traditional market research techniques used for measuring attitudes, brand images, and perceptions are not helpful in this situation. They do not provide the level of detail required for product or service planning in a QFD environment.

Even when companies make a concerted effort to inquire and listen to their customers, there are significant barriers to listening to the Voice of the Customer. Meeting customers face to face is the most obvious and direct approach, but it tends to be unreliable. One problem is "respondent bias", which occurs when the person being interviewed knows which company the interviewer represents. Another problem is the "rehearsal effect" which means that the interviewers very often do not hear the person they are interviewing because they are mentally framing or composing their next questions. Instead of listening to the customer many interviewers follow a natural tendency and attempt to solve their customers' problems. Finally, there is the problem with customers being unable to specify the engineering characteristics they want. The appeal of QFD is its ability to translate from the requirements customers really can tell you about into the engineering characteristics that they may not understand at all.

An important step to overcome the mentioned barriers is to do interviews anonymously. Contracting with a market research firm or other outside agency may be required to conduct to interviews. Depending on the respondents and the subject matter, group interviews may be more appropriate than one-on-one interviews. The stimulation of discussion and creative thinking is more achievable with groups.

Customer attributes must be analysed carefully before making design decisions. The Voice of the Customer should be organised into a hierarchy. The importance of each attribute at each level must be expressed so that trade-offs can be made.

---

[18] A more detailled description can be found in: Day 1993, p. 32; Klein 1990; Dahlheimer 1995

[19] Klein 1990

3.5 The VOCALYST process

The most common method utilized to systematically collect and structure the Voice of the Customer is the VOCALYST[20] process. This technique has been successfully applied in a wide range of both product and service industries[21]. The VOCALYST process consists of four steps:

VOCALYST Step 1. Customer interviews are the raw material for the Voice of the Customer. They can be group interviews or one-on-one discussions. Regardless of the format chosen, the interviews are tape recorded. This permits the interviewer to concentrate on the respondent rather than on trying to understand and absorb everything that is said during the interview. It is important for the interviewer to be ready to probe for the real Customer Attribute whenever an Engineering Characteristic is mentioned.

Attribute lists must be developed with direct customer input. It is important at this stage to contact as broad a cross-section of customers as possible. Intermediate customers as well as final consumers are all valid respondents.

VOCALYST Step 2. The tape recorded interviews are then transcribed for detailed analysis. Transcripts can be read in one-third the time it takes to listen to a tape. They allow many more people to be part of the process and to have direct access to the customer language. Some recent research at M.I.T. revealed that multiple analysts were necessary in order to identify a reasonably high percentage of the attributes[22]. These analysts use highlighters to mark the actual words and phrases in the transcript that begin to define the Voice of the Customer. Sometimes it is necessary to paraphrase the respondent to capture the idea behind a paragraph of words, although the goal is to stay as close as possible to the actual words the customer uses.

The highlighted phrases are then extracted for further processing. The first step is to eliminate the phrases that are duplicated or almost exact matches. In addition, not every phrase is a customer requirement. Phrases that define target values for attributes, subsystem planning concerns, styling and design concerns as well as product attributes should be sorted out[23].

Customers tend to reword needs, solutions, and problem concerns. They will frequently tell the interviewer what they want without saying *why* they want it. For instance, if a customer says he wants a *heavy toaster*, he probably means not a toaster having a high weight, but a device that stands stable on the counter. The determination of the *root want* in customers' statements is crucial to the success of the VOCALYST process. Only if the real underlying need of the customer is completely understood, a customer-driven product development process can be achieved.

The phrases derived from the customer survey are sometimes too long and have to be converted to the concise form. On the other hand, one-word statements don't provide sufficient information to detail the customers' want. Therefore they must eliminated or converted into a longer statement. Negative customer statements are not useful, because we are not keen to know what the customer does *not* want. The phrases must be converted into positive statements. Those that carry more than one voice in a phrase should be split into single requirements.

---

[20] VOCALYST: **V**oice **O**f the **C**ustomer **A**na**LY**sis **S**ystem **T**echnique

[21] Klein 1990

[22] xyz

[23] Day 1993, p. 56

In a review of the verbatims, it may be apparent that many people have said the same things in different words. Where possible, these should be consolidated into one phrase. Although the number of voices after the second step of the VOCALYST process varies with the scope of the project and the complexity of the product, the final list of customer voices can easily exceed 100 list items[24].

VOCALYST Step 3. A requirement list that is based on 25 or fewer voices is very manageable in the remaining planning and deployment process. Matrices between 25 and 50 voices can be managed, but the task becomes more and more onerous as the amount of input grows[25]. The fewer the voices, the easier it will be to work with the remaining requirements in the matrices. For that reason is it important to control the number of customer requirements.

The third step of the VOCALYST process is to bundle or group the voices into a structure of primary, secondary, and tertiary requirements. The best approach to limit the matrix size is to use the affinity diagram concept[26] which is described in chapter 4.2. This can be performed directly by the customer or by the QFD team. Although the data is more source-orientated when this task is conducted by the customer, it will usually undertaken by the team whose experience enables the process to be more time efficient.

In the affinity diagram process, all voices are arranged into natural groupings. After that, headings for every group are developed. After the group titles are completed, the next step is to see which of these groups can be merged into larger groups. The outcome of the process is a tree structure of customer requirements on a primary, secondary and tertiary level. If necessary, the tertiary items can be structured in more detail into fourth, fifth or higher levels. It is necessary for the consistency of the tree, that all voices on one level possess the same degree of detail.

When the tree is completed, the voices on the third level or any other level with a number of 25-50 requirements is extracted. These 25-50 requirements are the final requirements that will be used for instance in the House of Quality.

VOCALYST Step 4. The final task is to assess the importance of each tertiary requirement. This is done in a quantitative customer survey. The survey can also comprise a competitive evaluation. This task will be discussed in chapter 5.

The result of VOCALYST is a Voice of the Customer created by the customer. It is objective and not biased by prior beliefs or assumptions. A scientifically derived Voice of the Customer provides support for both quality enhancement and a variety of marketing research programs[27].

3.6 Summary of the Voice of the Customer Input Process

The described process is valid for most of QFD projects. The most important items of the voice of the customer input process can be summarised as follows:

- Quantitative Surveys with recording device for each customer group
- Transcriptions of records

[24] Klein 1990

[25] Day 1993, p. 175

[26] Day 1993, p. 177

[27] Klein 1990

- Marking and extraction of customer statements with different analysts
- Condensing requirements
    - Elimination of duplicate statements
    - Converting of one-word statements
    - Converting of negative statements
    - Splitting of multiple voices
    - Filtering solutions, characteristics etc.
    - Generating concise requirements from statements
- Group requirements into affinity diagram
- Develop tree diagram from affinity diagram
- Extract one level from the requirement tree structure
- Perform quantitative survey to assess:
    - Importance of requirements
    - Competitive situation percepted by the customer.

3.7 Computer Support for the VOC Input Process

The number of customers statements extracted from the transcripts can easily exceed 2000. In a QFD project of the University of New South Wales in Sydney, a customer survey to improve university courses was performed. The number of customer phrases from the two customer groups, *students* and *employers*, was 1391. This number can not be handled manually, computer support is required.

3.7.1 Provided Functions in the QFD-Support System

The *QFD-Support-System* prototype covers some functions of the necessary steps in the requirement generation process. These functions comprise the first two steps of the above described VOCALYST process:

- Importing text files with the survey transcripts into the system
- Marking and extracting customer phrases
- Merging different sources of customer voices into one database
- Eliminating duplicates
- Searching for one-word statements
- Splitting of multiple voices
- Converting of negative statements into positive statements
- Generating concise requirements

Using the system, an unstructured list of requirements can be generated. Functions concerning grouping and structuring the list and requirement assessment are not implemented.

3.7.2 Extraction of Voice of the Customer

After the qualitative survey has been conducted, the customer verbatim is entered into text files. Without the use of computers, these files were printed out and the phrases were marked with a text marker. The marked text would then be entered into a database. This task is time consuming and mistakes are inevitable. If the wrong text passages are marked, it is difficult to correct.

The prototype *QFD-Support-System* offers a VOC-Extraction-Module to perform this task more efficient. The basic task of the module is to extract customer voices from ASCII text files into a database table which carries certain additional information about every voice. This data is necessary to retrieve the source of a customer's requirement in later stages of the requirements generation process.

Basically, the transcript of the survey is split into sections which represent customer verbatims related to a certain question of the survey. Every statement is assigned a question number that represents a section of the survey. The main advantage is that the merged customer phrases can be classified and sorted by topic. This later becomes interesting in the requirements condensing process. For instance, whilst it is not easy within a set 1000 voices to find similar customer voices saying almost the same, the duplicate elimination process becomes a lot easier when the search has is performed within 10 topic-related sections of sub-groups of 100 voices.

All transcripts can be worked on by different analysts. Though their work is identical, the marked customer requirements will vary. Many customer voices will mean the same or be literally identical. A maximum of customer voices can be extracted. The identification of the analyst is stored with the customer phrase.

Every section of the transcript is analysed for customer demands. The phrases are marked with the mouse of the keyboard and copied into a list of voices of the customer. After the text is searched completely, the requirements are stored into the database table which also comprises also a *Customer ID*, the *Analyst ID*, the *Question No*, and the *Survey Date*.

Although the basic concept of the module will be maintained, the following improvements are suggested:

- Enlargement of the data structure. The information in the database table can be enlarged by the data fields *Customer Group* and *Comment*. The customer group information is necessary to provide special filters that can extract sub-groups in subsequent modules of the system (see chapter 3.7.4). A comment is useful for context information or further explanations of the analyst. The new data structure is shown in Figure 3-3.
- Flexible composing. It is not possible to compose requirements of separate text parts. That may be necessary if more than one requirement is contained in one sentence.
- Quick marking. The marking of the requirements has to be conducted very precisely. Therefore quick and efficient working is not possible. Markings should adjust themselves automatically to the beginning or end respectively of the words that they embrace.
- Coloured background of markings. In the module, markings are displayed with brackets only. This is confusing especially when requirements are defined over a longer text part.
- To take back markings, exactly the same text has to be marked again. This should be possible just clicking somewhere inside the marked text and pressing an *Unmark* button.
- The maximum size of an imported textfile is 30 Kbyte. For that reason, many textfiles have to be split and worked on separately. It should be possible to work on files with at least 100 Kbyte.
- To further facilitate work,"Drag and Drop" functions could be implemented.

Additionally to the suggested points, many parts of the implemented code had to be corrected or improved. A detailed description of these issues would force to focus on the Visual Basic programming language, which is beyond the scope of this thesis. All code related improvements will not be discussed in this paper.

AnalystID	CustomerID	Question	Statement	Date	Customer Group	Comment
long int.	long int.	text(15)	text(100)	date	integer	text(100)

Figure 3-3 - Data structure of Customer Statements

3.7.3 The Merging-Module

After the extraction the Voice of the Customer, the voices have to be condensed into a final list of customer requirements. To perform the condensation, all voices from different customers, customer groups and analysts have to be in a single database table to ensure a consistent collection of requirements.

Usually the voices of different customer groups are stored in different database tables, or in different files on different computers. Even the format of customer voices may vary from the proposed format in chapter 3.7.2, firstly because it was created on another system or secondly data from former QFD projects on different systems is used. To merge all different types of customer voices into one database a flexible merging function is required.

In the prototype this task is performed by the "Merging-Module”. The source of the voices can be any table in any accessable database, though the module is especially focused on the Statements tables from the VOC Extraction Module.

As described in chapter 3.7.2, some additional data is required for every extracted customer statement. The Analyst ID is required to follow back the original extractor of the customer phrase. A question number is required to assign a statement to the context of a section of the survey. Customer ID and Customer Group are necessary the filter out different customers or customer groups.

All customer voices will be converted into concise requirements in the following parts of the system. To be able to retrieve a requirements’ source, information about the source statement and its origin (question number, customer number, customer group) is needed. This information must not be changed during the process. For that reason, the source information is stored separately in a database table called *VOC Masterfile* that comprises all extracted customer voices. The actual converting from VOCs into Requirements is made in one or more "Requirement Workfiles”.

A requirement can have more than one source, although a source is not compulsory in the later process. This because requirements also have to be able to be generated out of the blue sky, which leads to a cardinality of (0,n). The cardinality of the customer voices is (0,n) as well. Voices can be converted into one or more final requirements, but this is not necessarily required. As described in chapter 2, masterfile and workfile are related in a many-to-many relationship.

The resulting conceptual data structure is illustrated in the Entity-Relationship diagram in Figure 3-4.

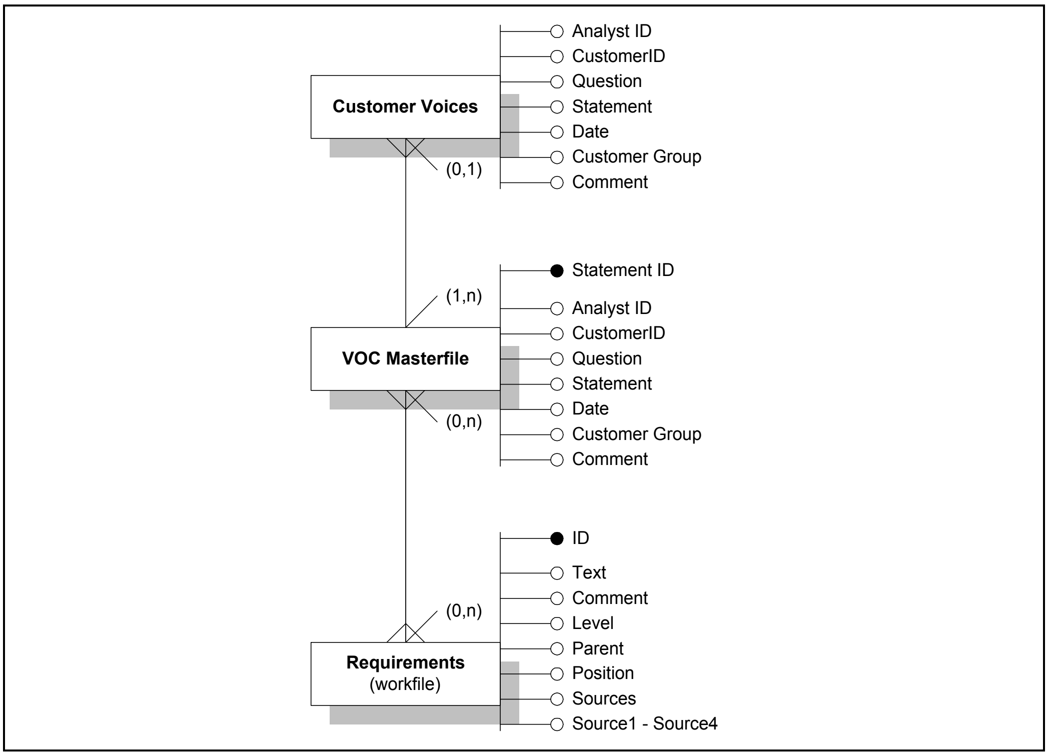


Figure 3-4 - ER Statements, Masterfile, Workfile

Kommentar: ER_State_VOC_Requ.vsd

The basic structure of the Merging Module in the prototype had to be redesigned completely. The following major changes were made:

- Introduction of the principle "One project - one database”. This means, that all QFD information related to a QFD project is stored in exactly one database. This has the following advantages:
    - Less confusion for the user. When switching through different databases and their tables, it is hard to maintain the overview. The problem increases, when many QFD team members work on the same system.
    - The database has to be chosen just once. A chosen database is the default for all operations until another database is selected.
    - Facilitated database management. Only one database has to be maintained and backed-up.
    - Facilitated software implementation.
    - The number of Masterfiles can be reduced to one (see below).
- One VOC Masterfile for the whole project. A serious drawback of the prototype occurs when Customer Statements are stored in more than one masterfile or even database. The corresponding Requirements table can not be merged with other Requirements tables at a stage later to obtain a final list of condensed requirements without the possibility of the source information being lost. The limitation to a single masterfile has additional advantages:

  - While serious data inconsistency and a system crash is likely when accidentally a wrong combination of masterfile and workfile is accidentally chosen, the can be avoided structurally.
  - The selection of the masterfile is no longer necessary because the masterfile is managed by the system.
  - The masterfile is invisible for the user.
  - Data inconsistency is avoided structurally.
  - Requirements tables can be merged or split without losing source information.
- Redundancy check. A user may accidentally attempts to merge a statements table that has already been merged to the masterfile. This leads to redundancy and inconsistency of the data. Therefore it will be checked before merging a statement if a record exists with exactly the same data in it.

3.7.4 The VOC Data Tools

The most important task of the VOC input process is the condensation and conversion of the VOC statements into customer requirements. In the prototype, this is implemented in the VOC Data Tools Module. The database table containing the customer statements is the workfile created in the Merging-Module.

As described above, the following sub-tasks have to be performed:

- Elimination of duplicate statements
- Conversion of one-word statements
- Conversion of negative statements
- Splitting of multiple voices.
- Filtering solutions, characteristics etc.
- Generating concise requirements from statements

Before the sub-tasks are described in more detail, some general aspects about the tool set will be considered.

The major problem of the condensing and converting task is the amount of data that is merged in the requirements workfile. Even in smaller QFD projects the extracted VOC statements can exceed 1,000 records. Larger projects may deal with as many as 10,000 records.

To diminish the impact of this problem, it is useful to create sub-sets of data. For that reason, every voice carries additional information as described in chapter 3.7.2. In the module, a filter can be set which extracts::

- statements related to one or more survey questions
- statements of one or more particular customers
- statements of one or more particular customer groups
- statements containing certain keywords
- statements extracted by a particular analyst

The filtered data can be condensed and converted separately. Unfortunately, an option for multiple filter criteria is not implemented in the prototype. Further, the data can not be filtered by customer groups. In the new system these drawbacks have been removed.

Further, the data can be sorted by different criteria in up to three hierarchical levels. The sources of requirements are retrievable. After the process the requirements can be printed out on cards.

The tool set of the prototype comprises functions for the elimination of duplicates for the conversion of one-word and negative statements, splitting of multiple voices and for the final conversion from customer voices to the concise customer requirements. The tools are basically satisafctory although many implementation faults had to be corrected.

4 The Affinity Process

4.1 Controlling the Number of Requirements

A set of up to 25 requirements is very manageable in the further planning and deployment process. Matrices with 25 to 50 requirements are still manageable, but the task becomes more and more onerous as the number of input grows. Requirement lists of more than 50 are not recommended[28]. Although the optimum number of requirements depends strongly on the particular project, 20-30 items can be considered a good compromise between grade of detail and complexity[29].

For instance if our final requirement list comprises 25 requirements and every item may generate 1.5 design characteristics, then we have to deal with 950 (38x25) relationships between requirements and characteristics and 703 co-relations between the requirements in the roof of the matrix. A constant ratio of requirements to characteristics presumed, the number of relationships and co-relationships raises quadratically with the number of requirements. The fewer the voices, the easier it will be to work with the remaining requirements in the matrices. As the matrix grows, some team members will show signs of fatigue and loss of interest. The consensus decision process will suffer and the overall value of the QFD process and its output will be diminished[30]. For that reason it is obviously important to limit the number of customer requirements.

The final list of requirements condensed by the VOC Data Tools Module usually comprises more than 100, or even 200 items when, they are derived from more than 1000 customer voices. Using the Affinity Diagram Process we try to find a way of further decreasing this number without diminishing the impact of the customers' voice.

4.2 The Affinity Diagram Process

The *affinity diagram* is one of the Seven New Quality Control Tools being proposed by Mizuno Shigero[31]. It is based on the *KJ method* developed and popularized by Kawakita Jiro.

According to Mizuno, "the KJ method clarifies important but unresolved problems by collecting verbal data from disorganized and confused situations and analyzing that data by mutual affinity.[...] Then the data areas are reduced to narrative rather than quantitative form". Using the affinity diagram, a large set of verbal data can be split into natural groups which are formed by the natural relations of the data. The process of the KJ-method is, when compared to other management tools, creative other than intellectual.

The affinity diagram process is conducted in five steps:

1. Collecting narrative data
2. Transferring narrative data onto cards
3. Sorting the cards into groups
4. Labelling the groups

---

[28] Day 1993, p. 175

[29] Terninko 1995, p. 78

[30] Day 1993, p. 176

[31] Mizuno 1988

5. Drawing the diagram

We will apply the affinity diagram for the organising of customer requirements. The first task is to write down all remaining customer voices onto cards.

The following task is the actual affinity diagram process. Cards containing similar items are grouped together on the basis of their affinity. This sorting is done not on the basis of reason, but on feeling. Mizuno stresses that the term "*feeling* refers to a state that precedes logical consciousness”[32]. What is desired is the *impression* that the cards group themselves. The cards are not to be gathered according to a certain classification scheme or on the basis of certain key words because they are not supposed to be *classified,* but simply to be grouped. After a few cards are in a group, the group can be labeled.

Having finished the grouping process, the grouped cards should be read one more time and checked to see if they are properly arranged. Inappropriately assigned cards are re-assigned or taken out of the group and returned to the presorting pile. The cards that appear to be properly grouped are given a label that represents the characteristic of the group. The label should convey the meaning of the cards completely, but not express more than the cards actually do. Abstract terms should be avoided. Ordinary expressions are more useful because they cannot lead to misunderstandings and do not allow interpretion of a meaning that is not intented by the content of the cards.

The affinity diagram chart is drawn when all groups are checked and labeled.

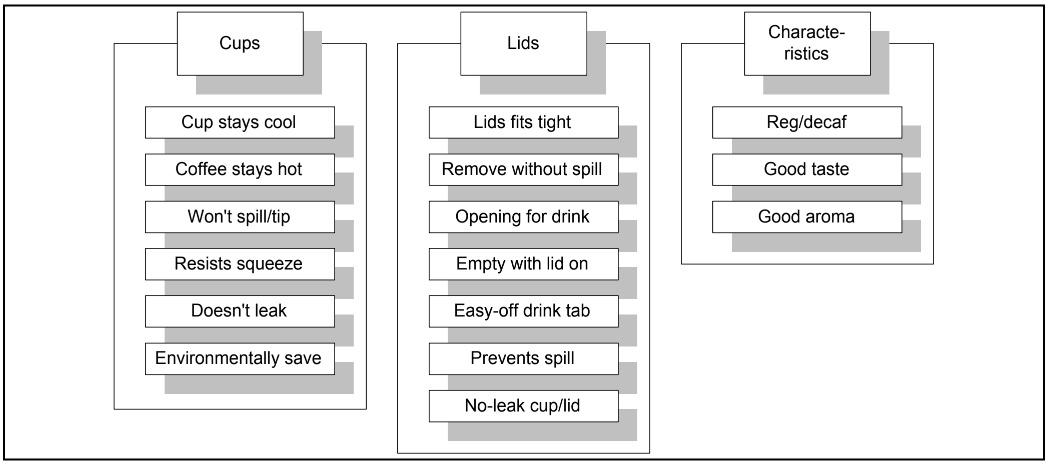


Figure 4-1 - Affinity diagram

Affinity diagrams are an efficient method of organising large amounts of unstructured verbal data. They enforce a high degree of organisation and screen hidden relationships between the items and item groups.

---

[32] Mizuno 1988, p. 126

4.3 Application of the Affinity Process for QFD

The third step of the VOCALYST process is to group the customer requirements into a structure of primary, secondary, and tertiary requirements. The best approach to the issue of limiting the matrix size is to use the affinity diagram concept[33].

The affinity process can be performed directly by the customer or by the QFD team. Although the data is more source-orientated when this task is conducted by the customer, it can be achieved much quicker when carried out by the team.

In the affinity diagram process, all voices are arranged into natural groupings. Then, headings for every group are developed. After the group titles are completed, the next step is to see which of these can be grouped into larger groups. The outcome of the process is a tree structure of customer requirements on a primary, secondary and tertiary level. If necessary, the tertiary items can be structured in more detail into forth, fifth or higher levels. It is necessary for the consistency of the tree that all voices on one level have the same degree of detail.

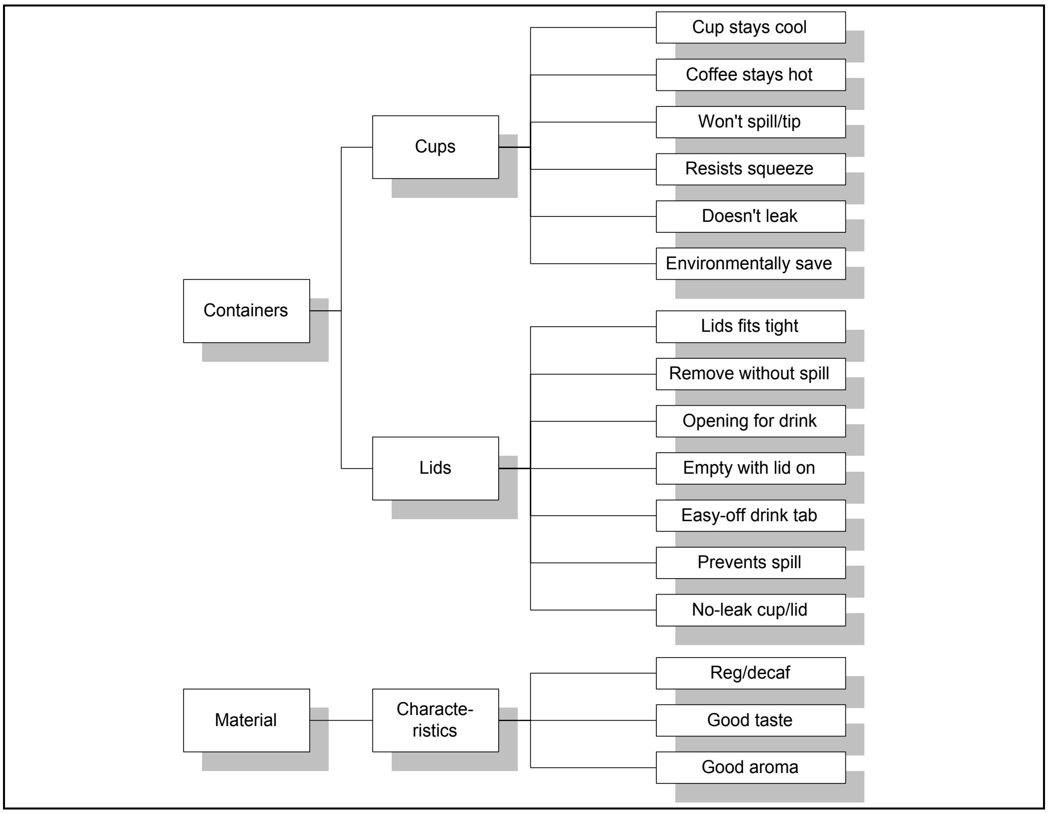


Figure 4-2 - Structured Requirements for a Coffee Cup

[33] Day 1993, p. 177

When the tree is completed, a certain set of requirements is extracted from the tree. These requirements form the final list of requirements that will be used for the qualitative survey, later in the Preplanning Chart or the House of Quality. There are two methods of extraction:

- Extraction by level. The tertiary requirements are extracted from the tree. This extraction method usually provides a reasonable number of requirements. A crucial prerequisite for this method is that all requirements of a particular level have the same degree of abstraction. This implies that tertiary requirements may be divided into more detailed sub-requirements without including these into the extracted requirements set.
- Exclusion of parents. Items of the tree structure that form a heading for more detailed requirements are called parents. Tf they are the last item on a branch they are called children. Following the extraction method *exclusion of parents*, only the children of the tree structure are used as final requirements.

4.4 Computer Support for the Affinity Process

4.4.1 Benefits of Computer Support

Westphal states that "it doesn't make much sense to do [the grouping job] with the computer, because it needs creativity and an overview of the requirements."[34]

He's terribly wrong.[35]

Although the Affinity Process is mainly a creative process, computer support can facilitate and increase efficiency of the following aspects:

- Input. The Affinity Process is conducted with cards containing customer requirements. These cards are usually created manually. When the data is already stored on a computer, the cards can be printed out, eg by using a database system or the VOC Data Tools described in chapter 3.7.4. Even this task would no longer be necessary if the affinity process could be completely conducted on the computer.
- Process. During the grouping process, similar requirements are grouped together. Doing so, requirements having similar meanings will be merged together or the duplicate will be put aside. It may be noticed that a requirement is missing in the structure. A new card must then be drawn. New headings have to be written on cards. It can be presumed that these tasks can be performed by means of a computer as successfully as using cards.
- Output. When the Affinity Diagram is ready, it has to be drawn on a sheet of paper and the structure of the data has to be entered again into the database. These tasks are inapplicable if the affinity process is conducted on the computer.
- Source Retrieval. The requirements written on cards usually have one or more customer verbatim sources. The voices of the requirements can not be retrieved easily during the affinity process. For that reason, their real meaning, that is, the underlying customer need, stated by the customer himself, is sometimes lost. Performing the affinity process on the computer, the sources of a requirement can be retrieved at any time by just pressing a key or clicking the mouse.
- Save / Recall. The affinity process is usually conducted as a single task. Once the process is commenced, it will be conducted straight to the end. For whatever reason, it may be

[34] Westphal 1996, p.128

[35] This isn't meant personally, Ingo!

important to stop the task at a certain stage. Perhaps further customer input may be necessary or perhaps some phrases were forgotten. To save the work and continue on another day is only possible if all cards are left on the table. Obviously, this leads to difficulties. Using the computer, the work could be saved or recalled by a mouse click.

- Change Management. QFD is an iterative process. Often the team realises that something should have been done at a preceding stage of the process. For instance, after the affinity diagram is completed, the team may decide the include another customer group in the QFD process. To change the affinity diagram manually is an onerous task, because the complete diagram has to be redrawn. On the computer, the task would take a fraction of time, even if including the changes in the database.

4.4.2 Maintaining the Overview

If an affinity diagram is displayed on a computer screen like it is grouped on a large table, it is hard to maintain the overview of all parts of the diagram. The Software "Memory Jogger Plus+"[36] and the HyQIS System follow this way.

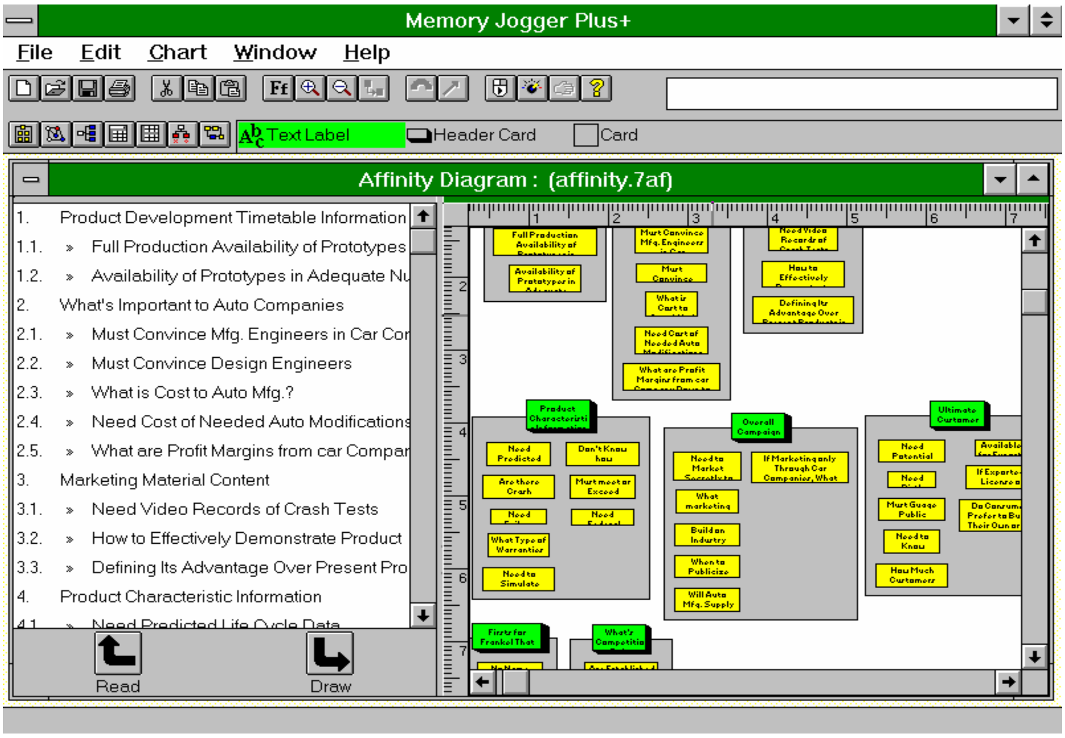


Figure 4-3 - Memory Jogger Plus - Affinity Sample

As can be seen in Figure 4-3 in a screen shot of the "Memory Jogger Plus+" with only 40 items, either the font of the cards has to be set very small or only a part the diagram can be shown at the same time. It is impossible to maintain the overview even with a small number of items. A

[36] Brassard 1989

grouping task with 200 items is virtually impossible to perform. For that reason, new ways of screening an affinity diagram have to be explored.

In this thesis it is suggested to display the content of the affinity diagram in a list that is divided into different groups. A screen shot of QFD-Manager's *Affinity-Module* of the is shown in Figure 4-4.

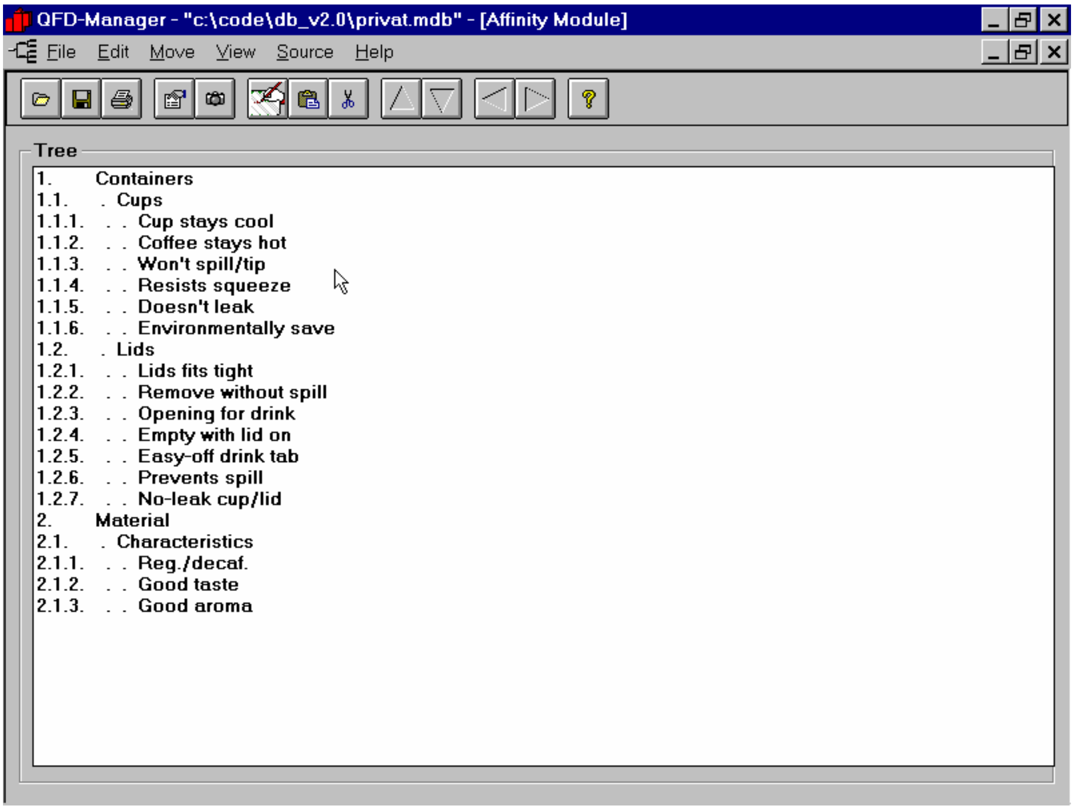


Figure 4-4 - QFD-Manager - Grouping Module

This list structure has many advantages:

- Optimal use of available screen space.
- Excellent overview.
- Optional automatic heading number assignment. Using this option, the overview can be kept very easily even when skipping through large amounts of data and list items.
- Grouping in more than one level possible. The grouping with the original affinity diagram can only be done in one level structure. There are only groups and cards. A group can not contain a sub-group. Using the list display, the grouping can be done in as many levels as wanted.
- One Tool for two tasks. After the actual affinity diagram is completed, the groups themselves are structured into a tree structure and the tertiary items are structured in more

detail. This can performed with a list without using a tree structure tool. The same advantages of the list apply when compared to a tree structure.

- Vertical scrolling only. A list is only scrolled up or down. This is easier for the user than to scroll in both directions which would apply in implementation as the *Memory Jogger*.
- Display like in House of Quality. The list structure of the requirements is identical to the display in the House of Quality or other QFD Matrices. The user is able to see the final work directly on the screen. An affinity diagram chart would first have to be transformed into the list structure.

4.4.3 The QFD-Manager Affinity Module

To realise the affinity process on the computer, an enlargement of the data structure of the requirement tables is required. The table has to contain the *Text* of the item, the *Level* of the item, its *Parent* (the item it is submitted to) and additionally its *Position* in the list. The position field is not only necessary to save the requirement structure but also to provide a saved storage of inconsistent lists. A list is inconsistent when one or more items have no parent. Without the *position* information, this list could not be saved. Like in the Data Tools, each item is identified by a unique *ID* number. Finally, each item should be able to carry a comment.

In Figure 4-5 the required data structure for the Affinity Module is summarized:

ID	Text	Comment	Level	Parent	Position
long int.	text(100)	text(100)	integer	long int.	integer

Figure 4-5 - Data structure for Affinity Module

The Affinity Module is not only designed to structure requirements, it is a rather universal and flexible module that can handle any kind of data, eg Design Characteristics, Functions, or Concepts. For instance, the outline of this thesis was designed by means of the Affinity Module[37]. Even if the data structure of the different entities varies, the only criteria for the usability in the Affinity Module is that the six necessary fields are provided as described above.

The screen of the Affinity Module consists basically of two windows. The first is called *Clipboard* and forms the presorting pile of the affinity process. The second window contains the affinity list or the *tree structure* respectively. When the module is started, all items from the chosen database table are initially on the clipboard unless they were previously grouped into a structure.

Before the affinity task starts, the user can create some initial group headings on the tree. The text of these headings is not important at this stage and can be changed later. The items on the clipboard can be moved to the groups on the tree. The process continues until all items are moved from the clipboard to the affinity list.

The text of items and headings can be changed, deleted or new ones can be created. If customer requirements are grouped, the original voice of the customer including all related source data can be retrieved. Items can be moved from one group to another or be arranged within their group. The level of an item can be selected and changed from one to five. The mouse can be used for all task including the user-friendly and very time-efficient *Drag and Drop* functionality.

---

[37] ...which might be not the be advertisement for the QFD-Manager.

Additionally, it is possible to move entire groups to another position within the tree structure. This is especially useful when dealing with large amounts of data on trees with a deep level structure. Another useful function is the increase or decrease of an entire group's level. This is useful if a whole group shall be shifted to another level. Further, there are some options that can facilitate special movement and grouping characteristics. All options can be saved so that they can be used during the next program session.

The level of the items is indicated by a tabulator and a preceding number. For instance, the number "2.3.4." would indicate the item is the fourth requirement in the third group of the second main group. A screen shot of the Affinity Module is displayed in Figure 4-4. After the first steps of the affinity task are conducted and the clipboard is empty, the tree structure can be maximized on the screen to utilize the full size of the screen for the detailed grouping. If both windows are to be displayed the user can select if the windows are displayed below each other (horizontally tiled) or next to each other (vertically tiled). For items with a short and concise text, the vertical tiling is advantageous. For long statements the horizontal tiling is better. To further adapt the display interface to the user, the font style and size of both windows can be selected by the user.

After the tree structure is completed, items can be extracted by the different extraction criteria described in chapter 4.3. The tree can be printed or saved as an ASCII text file. All changes on both clipboard and tree can be saved or recalled at any time.

5 Quantitative Survey

5.1 Importance Ratings

5.1.1 Quantifying Customer Requirements

After the final list of customer requirements is generated by the affinity process, the importance of every item has to be assessed. Although all customer wants are probably important, it is necessary to know their relative importance. Importance ratings play a key role in the QFD process. They help to:

- Serve as weighting factors for customer requirements
- Identify critical requirements (high-importance requirements)
- Serve as multipliers for other parts of QFD matrices.

No other part of the matrix has as much influence on the outcome of the process as the importance ratings[38]. Therefore, customer ratings must accurately reflect the customer's opinions. The effort and expense for satisfying high-importance requirements may be great. Therefore, the company should be sure that it really has identified the accurate ratings of customer needs. For that reason it is not sufficient to just estimate the importance of requirements. This form of internal evaluation is highly subjective and adds sources of variation from reality into the system[39]. The data has to be obtained directly by the customer. This is done by performing another customer survey, the *quantitative survey*.[40]

Like in the qualitative survey, appropriate methods comprise personal one-on-one interviews, group discussions and mail questionnaires. A comprehensive overview of survey methods can be found in Dahlheimer[41].

5.1.2 Ranking Systems

5.1.2.1 The Self-explicated Scale

In most QFD-projects, self-explicated or self-rated importance scales are used. They measure the customer percepted importance for each requirement independently from other requirements.

In the survey, a bi-polar interval scale is used[42]. The ends of the scales represent the minimum and maximum importance category, which carry the meaning "not important" and "most important". The in-between-levels can be left unworded or they can be assigned to statements like "average important". Instead of a worded description, the categories can also be identified by symbols. For a five-value scale, the symbols "--", "-", "o", "+", "++" appear to be useful to conduct the evaluation.

In recent QFD projects, the number of categories ranges usually from 5 to 9. Each category is assigned a numerical value which is used for subsequent matrix analysis and calculation, where the highest value equals the highest importance ranking. To describe the categories, only

[38] Guinta 1993, p. 52

[39] Gustafsson 1994, pp. 20-21

[40] Akao 1990

[41] Dahlheimer 1995, pp.142ff

[42] Dahlheimer 1995, p.128

integer numbers without decimals are used. Decimals can't improve the quality of ranking data, as proven in a study by Miller[43]. The experiment shows, that an individual cannot simultaneously compare more than 7 (+/-2) objects without being confused. Introducing decimals would increase the number of categories by a factor of 10.

When all survey data is gathered, the values for each requirement are collected and the average values are calculated. The *absolute importance* of the requirements is normalised to screen their *relative importance* in percent as depicted in Figure 5-1. It is important to emphasize, that the relative importance is derived from a non-relative customer evaluation.

Requirement	Imp.	Rel. Imp.
Cup stays cool	8	0.07
Coffee stays hot	7	0.06
Won't spill/tip	7	0.06
Resists squeeze	6	0.05
Doesn't leak	7	0.06
Environmentally save	6	0.05
Lids fits tight	7	0.06
Remove without spill	5	0.05
Opening for drink	8	0.07
Empty with lid on	7	0.06
Easy-off drink tab	6	0.05
Prevents spill	8	0.07
No-leak cup/lid	6	0.05
Reg./decaf.	8	0.07
Good taste	7	0.06
Good aroma	7	0.06

Figure 5-1 - Self-explicated Ranking

The advantage of the self-explicated scale is that it is easy to handle and easy to understand for the customer. The survey can be conducted in a short time.

On the other hand, the method has a severe disadvantage. The self-explicated scale does not provide *ratio-scale* data. That means, that some basic laws of mathematics can not be applied to the category values correctly. For example, category 6 is not necessarily twice as important as category 3. And the importance difference between category 1 and 3 may vary from the difference between 7 and 9. The importance values indentify primarily a certain importance category, and not a mathematically correct importance ratio between customer requirements. The quality of the self-explicated scale is often over-estimated and misunderstood.

Anyway, the self-explicated scale is still the most common used methodology in QFD projects because it is easy and time-efficient to perform. Despite the violation of the ratio-data feature, the method "still yields significant improvements in decision making".[44]

Griffin and Hauser have evaluated the validity of the most common ranking methods in a study of the Massachusetts Institute of Technology[45]. Their results are based on statistical data and

[43] Miller 1956, in: Dahlheimer 1995

[44] Terninko 1995, p.35

[45] Griffin 1992b

shows, that the self-explicated scale is one of the three best scales applicable for QFD projects, despite its ratio-data flaw.

5.1.2.2 Constant-Sum Measure

The constant-sum method asks the respondent for a *relative assessment* of the presented set of items. The most significant difference from the self-explicated method is that the customer is requested to make trade-offs between the customer requirements by distributing a certain amount points among the items. Therefore, it is a relative measure. Typically, the user is asked to allocate 100 points among the items.

Requirement	Imp.	Rel. Imp.
Cup stays cool	15	0.15
Coffee stays hot	2	0.02
Won't spill/tip	5	0.05
Resists squeeze	5	0.05
Doesn't leak	4	0.04
Environmentally save	10	0.10
Lids fits tight	5	0.05
Remove without spill	0	0.00
Opening for drink	3	0.03
Empty with lid on	5	0.05
Easy-off drink tab	8	0.08
Prevents spill	5	0.05
No-leak cup/lid	10	0.10
Reg./decaf.	8	0.08
Good taste	10	0.10
Good aroma	5	0.05
Total	100	1.00

Figure 5-2 - Constant-Sum Ranking

In contrast to the self-explicated scale, the constant-sum method provides a relative importance measure for the customer requirements. The method is more complicated. It is not so quick to perform for the respondent because the total sum of 100 has to be maintained. Still, it is an interesting alternative to the self-explicated ranking method. Terninko states that the results of the constant-sum ranking offer a "fairly close approximation of ratio data".[46]

5.1.2.3 Anchored Measure

Analogously to the constant-sum method, the anchored measure provides a relative assessment of the surveyed items. In the survey the respondent is asked for the most important requirement first. This requirement is assigned a fixed value, typically a value of ten. After that, the relative importance of the other requirements is evaluated by comparing them to the most important item. For instance a value of 5 would mean, that the requirement is half as important as the most important. All values from 0 to 10 are allowed. 10 means, that there are more than one requirements that share the degree of the highest importance.

[46] Terninko 1995, p. 36

The anchored measure method is a relative importance ranking method as well as the constant-sum method, and yields the same benefits. Its major advantage in comparison to the constant-sum scale, is that it is easier for the respondent to answer and more time-efficient method. On the other hand, the survey design is more complex and the measurement system demands a user's extended understanding.[47]

Requirement	Imp.	Rel. Imp.
Cup stays cool	10	0.14
Coffee stays hot	1	0.01
Won't spill/tip	2	0.03
Resists squeeze	5	0.07
Doesn't leak	2	0.03
Environmentally save	7	0.10
Lids fits tight	3	0.04
Remove without spill	0	0.00
Opening for drink	3	0.04
Empty with lid on	4	0.05
Easy-off drink tab	5	0.07
Prevents spill	4	0.05
No-leak cup/lid	10	0.14
Reg./decaf.	6	0.08
Good taste	8	0.11
Good aroma	3	0.04
Total	73	1.00

Figure 5-3 - Anchored Ranking

5.1.2.4 The Analytic Hierarchy Process

To overcome the ratio-data problem of the traditional ranking systems, some advanced techniques have been suggested in recent QFD literature. The most remarkable approach is the Analytic Hierarchy Process (AHP) proposed by the mathematician Saaty[48].

The AHP is conducted in a full pairwise comparison using a nine-point scale. For each possible pair of requirements, a comparison is conducted. The respondent chooses which requirement is more important than the other one. Then, he assign a *relative importance value* to the pair. Saaty proposes the following scale in which intermediate values can be used to identify the in-between-values.

---

[47] Urban 1993, p.264

[48] Saaty 1996, Saaty 1980

1:	equally important
3:	weakly more important
5:	strongly more important
7:	demonstratively more important
9:	absolutely more important

Figure 5-4 AHP Ranking Scale

After the survey process, the comparison data is collected in a matrix. Figure 5-5 displays an example for an analysis matrix. The values greater than one mean that the item in the row is more important than the items in the column. The cells below the diagonal represent the reciprocal cells above the diagonal. In the diagonal, a value of 1 is entered, because each requirement is equally important to itself.

	Market Size	**Cost to Support**	**Easy to Satisfy**	**Publicity**
Market Size	1.00	5.00	9.00	3.00
Cost to Support	0.20	1.00	2.00	0.50
Easy to Satisfy	0.11	0.50	1.00	0.33
Publicity	0.33	2.00	3.00	1.00
Total	1.64	8.50	15.00	4.83

Figure 5-5 - AHP Evaluation Matrix[49]

After this step the columns are normalised to 1.00, as depicted in Figure 5-6. The Row Sum of the normalised matrix values ("Total") are normalised as well, which leads to the final importance results ("Average").

	Market Size	**Cost to Support**	**Easy to Satisfy**	**Publicity**	**Total**	**Average**
Market Size	0.608	0.588	0.600	0.621	2.417	0.604
Cost to Support	0.122	0.118	0.133	0.103	0.476	0.119
Easy to Satisfy	0.068	0.059	0.067	0.069	0.262	0.066
Publicity	0.203	0.235	0.200	0.207	0.845	0.211
Total	1.00	1.00	1.00	1.00	4.000	1.000

Figure 5-6 - Normalised AHP Evaluation Matrix

[49] Terninko 1995, p. 40

The AHP Process yields mathematically correct ratio data, as proved by Saaty.[50] Another advantage of the AHP is that two-way comparisons are easiest for the human brain to perform.[51]

On the other hand, the customer may tire of comparing all pairs of combinations for the list of items to be ranked. Further, the AHP can only be conducted for a maximum of 7+/-2 items.[52] The reason is obvious; each requirement has to be compared with every other. For 9 requirements to be surveyed, this would lead to a number of 64 comparisons. Supposing that an average QFD project has 25 customer requirements, a full-comparison method is not feasible; 25 items would lead to 24*24=576 comparisons.

Although AHP has strong advantages, it is not feasible for most QFD projects because of its massive time consumption.

5.1.2.5 Cascaded Analytic Hierarchy Process

In most of the QFD projects, the team uses the extracted level-three requirements as input for quantitative survey. Each of these low level requirements (called children) are submitted to a requirement on a higher level (called parent).

The underlying structure makes it possible to apply the method of *cascaded ranking* to the AHP. Using cascaded ranking, the number of necessary comparisons can be reduced effectively. The time-effort problem which limits the practical use of AHP can be overcome. The central difference to common AHP is to perform the comparison process only in one branch of the tree at a time.

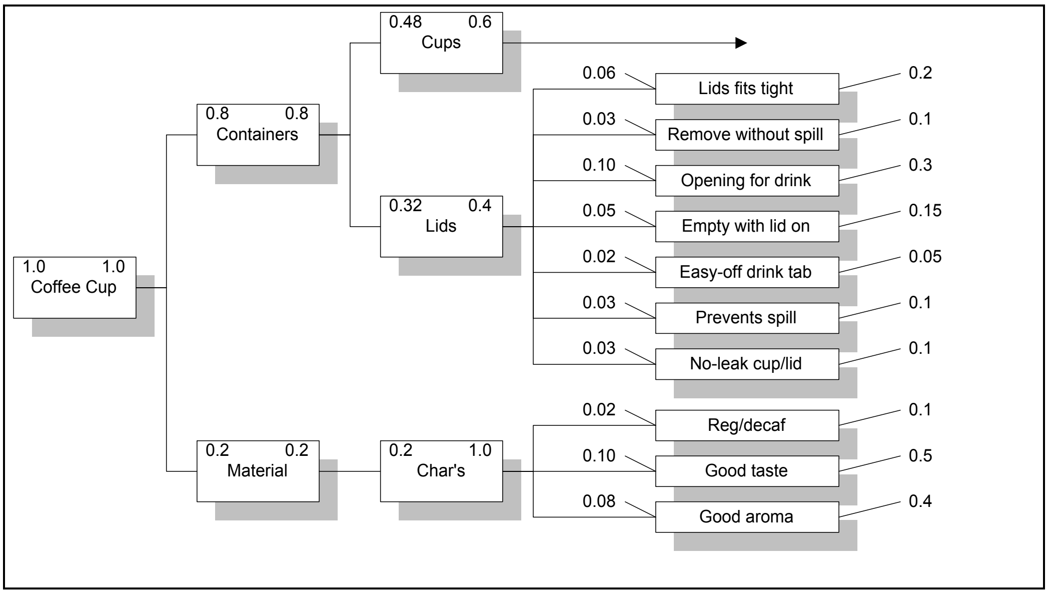


Figure 5-7 - Cascaded AHP

[50] Saaty 1996

[51] Terninko 1995, p.36

[52] Terninko 1995, p.38

As shown in Figure 5-7, the comparison is performed in the first step for the class-one-requirements (Containers and Material). The relative importances in the example are 0.8 and 0.2. In the next step, Cups and Lids are compared, which are ranking as 0.6 and 0.4 relatively to each other. The next branches are the groups "Cups" (6 items, not illustrated) and "Lids" (7 requirements). The necessary 36 (6*6) comparisons can still be managed with the AHP in a *finite time*.

In the second main group ("Material"), there is only one class-two-requirement, so no comparison is necessary on the second level. The relative importance value for Characteristics is 1.0. Only its children are compared to each other.

The relative importance value of an item within one branch is called *local importance*. In Figure 5-7, the local importance is displayed on the top right of the requirements. The sum of the local importances within one group is always 1. The *global importance* represents the importance of the requirement relative to all other requirements on the same level. They are depicted on the top left. The *global importance values* for the items on the branches of the lower levels are calculated by multiplying their *local importance* with all importance values of their superior items. For instance the requirement "Lids fits tight" is evaluated to 0.8 * 0.4 * 0.2 = 0.064.

The use of cascaded ranking implies the danger that wrong weights on a high level are consistently multiplied through the hierarchy and lead to wrong emphasis on the lowest level.[53] Additionally, a flawn structure of the requirements tree leads to incorrect and uninterpretable results. The hierarchical structure of the tree is no longer only means for structurizing the customer requirement information, it has a strong influence on the results of the QFD process. Using cascaded ranking, it is even more important that the requirements on each level have the same degree of detail.

Further, the survey structure generated by cascaded ranking is complex. The respondent has to assess the relative importance by conducting a full pairwise comparison for each group on every level. Although the number of comparisons is still reduced in comparison to common AHP, the effort is still considerable. In the example in Figure 5-7, the number of necessary comparisons is 69. The design of the survey becomes more difficult, and the evaluation process is more sophisticated.

On the other hand, cascaded ranking allows to deploy importance ranking *consistently* down to the lowest level. The gained data is indeed ranking data and mathematically correct.

Richard Zultner, as one of the most influencial researchers on QFD, highly recommends the use of cascaded AHP in QFD and claims that it delivers more accurate importance rankings than other methods.[54] Still, the main drawback is the number of comparisons and the high effort to conduct the survey.

5.1.3 Critical Customer Requirements

After the survey data is collected, the importance values for each requirement are calculated and normalised. In some of the presented ranking methods, normalised values are already gained by the process itself.

The requirements with the highest importance values represent the *critical customer requirements*. These requirements have to be considered first in the further development

---

[53] Griffin 1992b, in: Dahlheimer, p. 127

[54] Zultner 1993

process. Customers will only be satisfied when these requirements are met.[55] Although fulfilling other requirements can be also important, their impact on the overall customer satisfaction will be low in comparison to the critical requirements.

After the importance ranking is finished, the last step of the VOCALYST process is accomplished. The importance values are the first important pieces of information for the preplanning chart, which will be discussed in chapter 6.1.

5.2 Customer Competitive Assessment

The weighted list of requirements generated in the VOCALYST process has captured the voice of the customer and defines the customer's perception of the product. In the next step of QFD, this perception is used as the basis for a comparison between the company and its competitors. The Customer Competitive Assessment verifies the present market position of the company's current product. The Customer Competitive Assessment enables the company to: [56]

- Discover weaknesses in the fulfillment of critical customer requirements
- Detect reasons for the success of the competitor's products
- Identify the customer's perception of the company's product in comparison with the competitor's products
- Identify weaknesses in the competitor's product as a basis for new market opportunities
- Provide additional quantitative data for the preplanning chart[57] helping to further analyse the weight of customer requirements.

The customer competitive data helps to determine if a product will sell or not. Because of the value of this information and its power to make informed business decisions, the data has to have a very high level of accuracy. Its quality is crucial to the company's success and should therefore be obtained directly by the customer. Although marketing data takes time to collect and incurs expenses, taking shortcuts or omitting this step reduces the reliability of information on which critical business decisions are based.

The first step of the survey preparation is to identify which products will be used for comparison. The QFD team selects competitors who are successfully offering a similar product. The number of competitors varies. In some cases, the company competes with only one main competitor. In other cases there is a greater variety of competitive products on the market. Usually, up to four competitors are selected.

The scales and ranking methods for the competitive assessment are similar to the importance ranking systems, while the self-explicated scale from 1 to 5 is the most widely-spread. In the survey, symbols can help to overcome the drawbacks of numerical values as in the importance ranking survey (see chapter 5.1.2.1). All ranking methods described in chapters 5.1.2.1 to 5.1.2.5 can be applied for the competitive assessment analogously.

Due to the similarity of both importance ranking and customer competitive assessment, the surveys can be conducted at the same time. The survey input are the customer requirements on the one hand, and the competing products on the other hand.

---

[55] Guinta 1993, p.59

[56] Day 1993, p. 43

[57] see chapter 6.1

The output of the importance evaluation is a one-dimensional list of importance values. The output of the competitive assessment is a two-dimensional matrix, whose rows form the customer requirements and columns represent the competing products (see Figure 5-8).

		Competitive Assessment	
Requirement	Imp.	Company	Competitor
Cup stays cool	8	1.9	3.2
Coffee stays hot	7	2.5	3.8
Won't spill/tip	7	1.8	2.9
Resists squeeze	6	4.1	2.7
Doesn't leak	7	4.2	3.3
Environmentally save	6	3.8	1.7

Figure 5-8 - Customer Competitive Assessment

5.3 Computer Support for the Quantitative Survey

In the following sections, effective ways of conducting the qualitative survey with computer support will be discussed.

Among the five presented ranking systems, the self-explicated scale is the widest spread. The other ranking systems suffer from a low acceptance of the interviewees, because they have to be understood first, and it takes longer to answer to the questions. The AHP especially is not excepted as a user-friendly survey method. Although the gained data is of higher quality, the traditional self-explicated scale will be used for the interviews.

The traditional method of surveying requirement importances and customer competitive values is to conduct personal interviews or to mail questionnaires to the customer. All required data for the survey is already provided on the computer system, that is, the final list of customer requirements generated with the Affinity Module. An attempt has been made to find a "shortcut" for issueing and collecting survey data while fostering organization and reducing manual work.

For supporting the survey with computers, four different methods appear to be possible:

- Print out a questionnaire using the data already provided on the computer system.
- Export the data into a textfile, edit the text with a word processor and print out a questionnaire.
- Create a diskette with a survey software that is given to the customer. The diskette is returned after the customer has performed the survey. The surveyed data is stored directly on the disk and can be read by the QFD-System.
- Performe the survey online on a commonly used Wide-Area Network, eg the Internet.

The advantages and drawbacks of these methods are depicted in Figure 5-9. They are discussed in greater detail in the following chapters.

Medium	Directly printed Questionnaire	Questionnaire printed from exported ASCII file	Diskette	Internet
Required computer facilities at customer's site	none	none	IBM PC with Windows 3.1 or higher	Any computer with internet connection, Internet-Browser
Degree of automation for creating media	automatic	partly	full	full
Media flexibility	average	high	depends on survey software	average
Media distribution	mail or fax	mail or fax	mail	network
Degree of automation retrieving data	none (keyboard)	none (keyboard)	full	full
Retrieval speed	mail (3-5 days) fax (0-1 day)	mail (3-5 days) fax (0-1 day)	mail (3-5 days)	online (0-1 day)
Estimated customer acceptance	average	average	high	high

Figure 5-9 - Criteria for Survey Media

5.3.1 Printed Questionnaire

Printing out a questionnaire is the easiest way of conducting the survey. At the customer's site the only thing required is a pencil to mark the ratings for the requirements and the competitive survey.

The survey preparation can be performed by printing the questionnaire directly from the QFD software, or indirectly by exporting an ASCII file which contains the survey items to a text processor. Using the second methodology, further manual text processing work is required to adapt the raw text data to a well-formatted questionnaire. On the other hand, this additional work increases greatly the flexibility of the questionnaire. The company's logo, special introductions, notes or legends can be included in the questionnaire. The text can be formatted, designed and customised as desired. Using the QFD-System in reality, the author of this thesis supposes the flexible text file to be the more likely way utilized by the surveying companies.

A drawback of questionnaires is that the data of the retrieved sheets has to be entered manually into the computer system. This can be an onerous task, especially when dealing with many retrieved survey forms and items.

The customer acceptance can be estimated as being average. The customer at least should know know how to deal with printed questionnaires.

5.3.2 Diskette Survey

The QFD-Manager creates a diskette survey by storing a precedingly prepared evaluation program on a floppy disk, which is mailed to the customer. The customer can run the evaluation program directly from the diskette. The data entered by the customer is stored on the disk and the diskette is returned to the surveying company. Back at the company's site, the data can be read by the QFD-Manager automatically.

This unique survey method has to advantage of the full automation of the data flow. If all data is provided, generated or stored by a computer system, manual input is completely unnecessary.

The only required equipment at the customer's site to perform the survey is an IBM compatible PC with Windows 3.1 or higher. The customer's acceptance of this survey method can be supposed as being high, because it is a new and interesting method. A requirement for the acceptance of this method is that the software is easy to run and use. The user has to realise directly what to do without having to learn about the use of the application first.

A drawback is the low degree of flexibility. The survey can only be performed in the way the evaluation program is developed. Variations of this method have to implemented into the code of the evaluation program.

5.3.3 Internet Survey

The Internet is one of the strongest growing markets of the world. The Internet's growing popularity is mainly caused by the World-Wide-Web (WWW) introduced in 1990 and the Internet's electronic mail (Email) service.

The WWW is a hypertext-based information system. It consists of *pages* containing text, graphics and links. The definition of a WWW-page, the *HTML*-format (Hypertext Markup Language) is used. To use the WWW, the computer has to be connected to the Internet and has to have a *WWW-browser* installed . A WWW-Browser, eg Netscape[58], is a software that reads the content of WWW-pages from the Internet and displays the page on the monitor.

A page can contain certain *links*, which can be selected by the user. A selected link brings up another document which appears on the screen as soon as it is loaded over the network[59]. These links are called *hyperlinks*.

IBM Corporation states in their WWW pages about the history of the Internet[60]: "Nobody can say precisely how many people are using the Internet today, but there are estimated to be more than three million host computers with as many as 30 million users around the world.The number of users is growing by 15 percent per month. Today 78 countries have full Internet access connections. and 146 countries can exchange e-mail. Every 30 minutes, a new network signs on to the Internet."

Tony Rutkowski of General Magic Corporation: During the years 1990 - 1995, the World-Wide-Web increased its data traffic each year by more than 100%. The development of the number of hosts from 1989 to 1996 and the projected values for the rest of this century are depicted in Figure 5-10 and Figure 5-11.

---

[58] Netscape is a trademark of Netscape Corp., USA

[59] Further information about the WWW can be obtained eg from http://www.w3.org

[60] http://www.ibm.com/Features/ancient.html

Figure 5-10 - Internet Hosts 1989-1996[61]

Kommentar: I_Hosts1.bmp

Figure 5-11 - Internet Hosts - Overall Trend[62]

Kommentar: I_Hosts2.bmp

To create an internet survey, a WWW-page has to be developed that is capable of sending data back to the WWW server. This can be accomplished by the use of *forms*. A form is a WWW-page that contains certain *input fields* in that the user can enter text or values. After all values are entered, the form can be submitted to the surveying company via email. The emails containing the survey data can be retrieved over the network automatically.

Another solution for direct data transfer back to the server is the use of the Common Gateway Interface (CGI). Using CGI, the server executes a local program when receiving a form. The program is fed with the submitted data of the form. The data can be evaluated online and directly stored in a database and a response can be sent to the web user instantly. This solution requires more efforts to implement the evaluation program, but the retrieval of emails can be avoided.

---

[61] http://www.genmagic.com/Internet/Trends/slide-3.html

[62] http://www.genmagic.com/Internet/Trends/slide-4.html

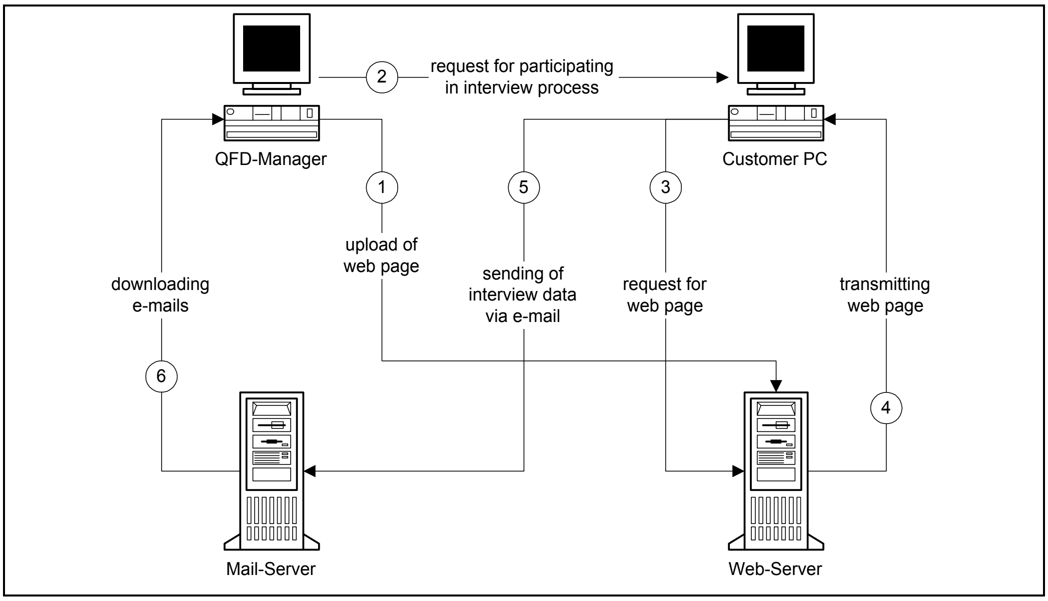
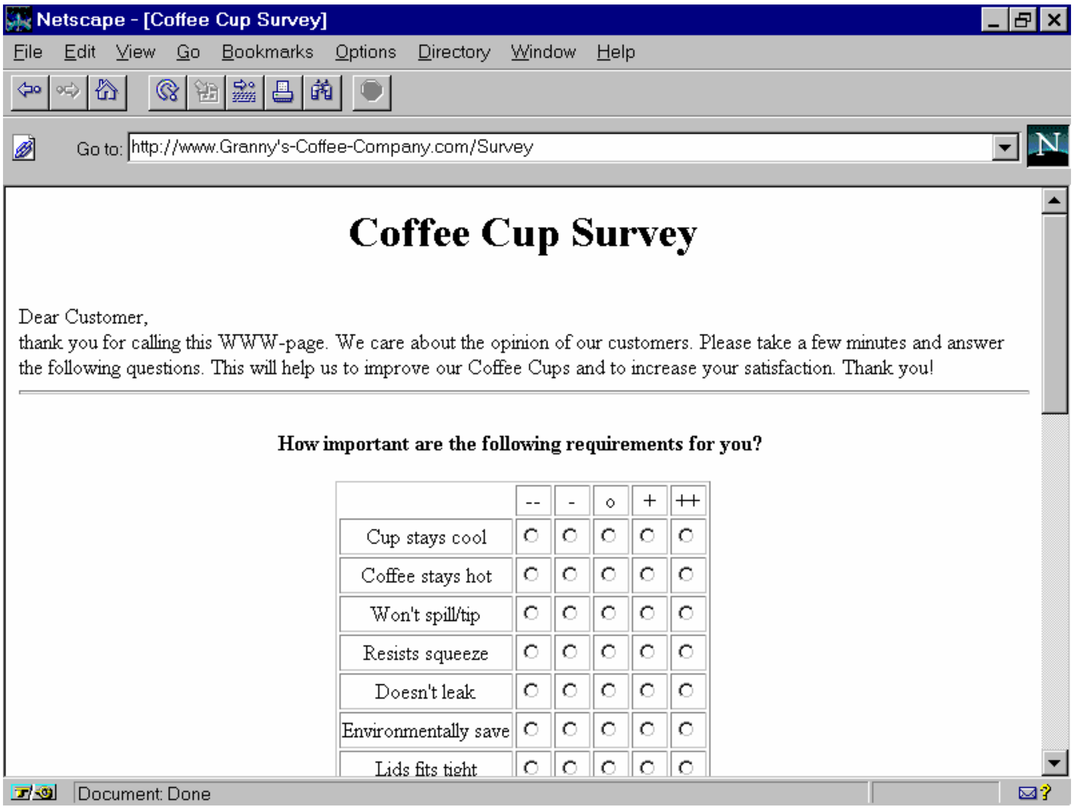


Figure 5-12 - Dataflow in the QFD-Manager

In the QFD-Manager, the first solution is implemented (see Figure 5-12). Each time a user submits a form with QFD survey data, an email is sent to a dedicated mailbox for the QFD-Manager. The emails can be automatically retrieved from QFD-Manager and stored into the project database.

As more users go "on line" via Internet, the customer base that is able to access the world wide web grows. The surveyed customer does not have to be a computer specialist, neither the user of the QFD-Manager. The WWW-page can easily set up by storing the page definition into a directory on the Server that is dedicated as WWW source directory. To fill out the form, the customer has to open just the WWW location and enter his opinion. Figure 5-13 shows a sample page of the QFD-Manager's Internet Survey using the Netscape™ Browser.

Among the presented ways to perform the quantitative QFD survey, the internet survey is the most automated. Neither paper, diskettes, nor any other materials are used. The information flow and data processing is exclusively performed on computers and the network. The customer only requires a computer with a WWW-Browser to handle the survey form and post it to the mailbox of the surveying company. Mailing or faxing questionnaires becomes as unnecessary as posting diskettes when using the Internet.



Kommentar: Html survey.bmp

Figure 5-13 - Internet Survey displayed by a Web-Broweser

The internet survey is the fastest way to retrieve the survey data back to the QFD system. After the customer is requested (eg via email) to have a look at the questionnaire form on the network, the survey data is posted immediately back to the mailbox of the surveying company. The retrieving time basically depends on the frequency the customer checks his emails containing the request to participate in the survey.

The flexibility of the survey depends on the QFD-System. It is possible to create a flexible WWW-page generation module with different options to include introductions and help texts for the customer. In the QFD-Manager, all these options are realised. If further modification is required, the page can be modified easily with a text editor after it is created.

The internet survey capability of the QFD-Manager is unique. Although some vague thoughts about the use of the Internet for QFD applications can be found in current literature[63], the presented solution in the QFD-Manager is more flexible and powerful.

5.4 Data Structures for Computer Assisted Surveys

The implement software for supporting the quantitative survey, certain data structures are required to properly create survey media and gather survey data.

[63] Stauffer 1996, p. 573

To create a survey, we need information about:

- The filename of the project database.
- The name of the (requirements) database table as a source for the items being surveyed.
- Which type of survey shall be performed (importance survey, competitive analysis, or both)
- The field name of an importance field in the source table, if importances are evaluated.
- The names of the competing products and the related field names of the product fields in the source table, if competitive analysis is conducted.
- Special options

In the QFD-Manager, a set of this information is refered to as *Survey Definition.* Survey Definitions are stored as records in the database table *Survey Definition Table*, which contains all survey definitions of the current QFD project. The Survey Definition Table is a unique system table in the project database that cannot be accessed by the user directly. The definition of this table is depicted in Figure 5-14.

ID	Survey Name	Table Name	Survey Type	Importance Field	Product Field 1	Product Name 1	Product Field x...	Product Name x...
long integer	text	text	integer	text	text	text	text	text

Figure 5-14 - Data Structure of Survey Definition Table

The field *ID* contains a unique identification number for every survey, followed by a name for the survey (*Survey Name*), which must be unique as well. *Table Name* contains the name of the source table within the project database. The name of the database does not have to be provided as it is already given by the location of the Survey Definition Table. Renaming the project database would lead to an error and necessitate redefining the survey. *Survey Type* contains a numerical value indicating which surveys shall be performed. Valid values are one, two and three, meaning "importance evaluation”, "competitive assessment” and "both” respectively. *Importance Field* contains the name of the database table field in which the final importance value for the requirements will be stored. Although it could be also defined later, it is advantageous to provide this information in early stages of the survey. Only in this case it can be guaranteed that the retrieved survey information reaches its target properly and that confusions are avoided. The fields *Product Name* and *Product Field* contain the name and table field for the products. Up to four products are allowed.

After a survey is defined, different media can be created on this survey definition:

- Questionnaires directly printed from the QFD-Manager’s Survey Module.
- ASCII files, containing the survey text. These files can be edited with any ASCII based text editor.
- Diskettes containing the evaluation program and a datafile for data being retrieved.
- Internet-Pages that can be transfered to or stored directly on a WWW-Server.

For collecting the retrieved survey data, another data structure is required. The collected data cannot be stored directly into the source table because the average values for the importance and product assessment must first be calculated. To guarantee a safe data gathering, a new database table called *Survey Data Table* is being used which collects all retrieved survey data in the QFD project. A record in the table represents a single customer vote for a particular requirement. It contains information about the importance of the requirement and the degree of fulfillment provided by the competing products. The data structure for the Survey Data Table is depicted in Figure 5-15.

ID	SurveyID	Customer Name	Date/ Time	Record ID	Importance	Product 1	Product x...
long integer	long integer	text	text	long integer	numeric	numeric	numeric

Figure 5-15 - Data Structure of the Survey Data Table

The actual survey data is carried in the fields *Importance, Product 1, Product 2, etc.* Additionally, information about the survey definition is provided (*SurveyID*). The customer name and the date of the survey performance are stored in the fields C*ustomer Name* and *Date/Time*. The field *RecordID* contains the identification number of the requirement that the data is related to.

For retrieving the survey data, the following functions are provided:

- Manual data input.
- Reading of data from survey diskette.
- Reading of posted internet forms from network mail server.

Although the underlying sub-procedures in the Internet-Retrieval are sophisticated, the user only selects the logon information for the Email account and presses the *Retrieve* button. All subsequent steps are performed automatically:

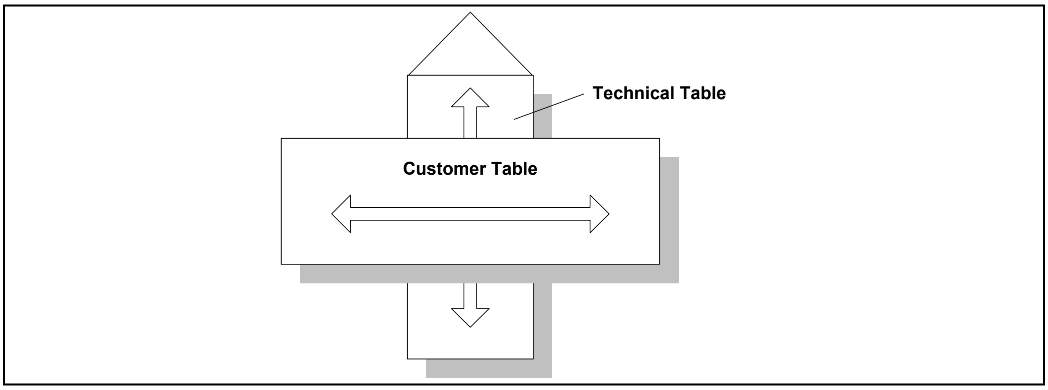
- Connect to POP3-Server,
- read messages on server,
- identify survey forms for the respective survey,
- download data from server,
- store data into database,
- delete mail from server.

6 Charts in QFD

6.1 Charts and Matrices

The House of Quality has two major components: one captures the customer information and one contains technical information as illustrated in Figure 6-1.

The Customer Table, also called *pre-planning chart*, captures the information related to the customers' wants, needs and evaluations. The Technical Table contains informations related to the company's determination of important technical issues and their amplification. Together, these two tables produce the main frame of the House of Quality.



Kommentar: charthoq.vsd

Figure 6-1 - Customer Table and Technical Table

In this thesis, tables used in QFD-projects are called *charts* to avoid confusion with the term of *database tables*. For instance, the Customer Table and Technical Table in Figure 6-1 are charts.

A *matrix* is a combination of two charts. One chart is drawn horizontally, the other vertically. The intersection of the charts contains the relationship values between the charts' items, which forms the core of the matrix. The House of Quality is an example for a matrix.

Matrices can occur in a variety of instances in QFD-projects. The content of the tables in the matrices is obviously different, but the purpose of each matrix is basically the same: The matrix describes:

- relationships between the items of the two tables, and
- provides special means of data transfer from one table to another.

Before matrices are discussed in greater detail (see chapter 8), charts as the main components of matrices will be focused on.

6.2 The need for a Pre-planning Chart

To get a bit more familiar with QFD-charts, the most important chart of every QFD project and its calculation will be discussed in more detail. This chart is called the *pre-planning chart* and represents the customer table, that is used as the horizontal part in the House of Quality matrix.

In chapter 5 the conduction of the quantitative survey was discussed which provides the importances of requirements. These rankings describe the importance of each want relative to the other wants of the list rated by the customer's view.

There are circumstances when it is useful to include more factors or criteria in the prioritisation process than just the customer importance ratings in order to reflect organisational goals or challenges. The pre-planning chart is a tool that can incorporate these other criteria. This matrix can combine multiple criteria such as customer importance ratings, marketing focus areas, competitive performance data and other aspects. The result is a composite importance rating that provides a key input for determining the weight of customer wants. In the following, this value will be called *requirement weight* or *row weight*. This value is one of the key values in the House of Quality. It is used to prioritise requirements and to allocate resources for these major issues if not every single customer want can be fulfilled. It is further used as a weight of a requirement to be fulfilled by technical design characteristics. The importance of design characteristics is strongly influenced by the requirement weight.

Many suggestions have been made to define an appropriate set of criteria for the pre-planning matrix to calculate row weights. The GOAL/QPC methodology[64] presents a process for combining several criteria as an absolute weight as a part of the development of the House of Quality. This methodology incorporates customer rated importance, the competitive gap of the company's product in comparison to the main competitors, and market positioning (sales points). The absolute weight is a calculated importance ranking for each want which incorporates all of these criteria. Customer wants can be prioritised based on these absolute weights.

Day suggests a different set of criteria for the pre-planning chart.[65] Beside the customer customer rating and the competitive values, he uses of the number of complaints concerning a particular customer want. He states that only five percent of the customer problems and complaints actually reach the manufacturer of the product. From the statistical view, for every customer complaint received by the company there are nineteen other customers similarly dissatisfied. Thus, the written or verbal complaints constitute major negative exclamation marks for the customer want in question[66]. Day also uses a *Salespoint* field and a text field named *Action* which can contain a short description of further QFD team investigations or steps.

The variety of different criteria used in pre-planning charts is immense and varies in every QFD project. Virtually any criteria selected by management or the QFD team could be of use in selecting the customer wants to be prioritised.[67] The calculation method of the final row weights can also vary in every chart and QFD application.

Thus, in this thesis a suggestion is made to cover all possible criteria that can occur in QFD projects. The proposed solution is not only capable to design highly flexible pre-planning charts, but also any charts with attached data being used in QFD matrices, such as the technical table, concepts or functions.

Before describing the detailed concept that is implemented in the QFD-Manager, some different ways of calculating the row weights of charts will be discussed.

---

[64] King 1989

[65] Day 1993, p.54

[66] Goodman 1989, pp. 37-40, in: Day 1993, p.54

[67] Yoder 1994, p.214

6.3 Calculation Methods for Requirement Weights

To calculate the absolute and relative requirements weights, there is a large variety of feasible calculation methods. Three basic types can be extracted from this large variety[68]:

- Multiplication of Scores
- Multiplication of Ratio Scores
- Addition of Scores

6.3.1 Multiplication of Scores

The traditional way to calculate the row weights is the Multiplication of Scores Method. The composite absolute row weight for each customer want is defined as the product of all of the criteria values. The relative row weight is the defined as the normalised absolute row weight. An example for this method is depicted in Figure 6-2. In the example, *Customer Importance* is the value ranging from 1 to 9 that is obtained in the quantitative survey directly by the customer. The *Competitive Gap* can carry the values 1, 3, 5 and 9 depending on the strength of the competitors product. This value is deducted by the competitive analysis. Finally, the *Sales Point* describes the impact that the fulfillment of the particular customer want will presumably have on the sale of the product. In the example, values from 1.0 (weak impact), 1.2, or 1.5 (strong impact) can be assigned.

The absolute row weight is the product of *Customer Importance*, *Competitive Gap*, and *Sales Point*, the relative row weight is the normalised absolute weight.

		Customer Importance	Competitive Gap (1,3,5,9)	Sales Point (1.0,1.2,1.5)	Abs. Weight	Rel. Weight
Work well on laundry	Grip things tightly	9.2	9.0	1.5	124.2	59%
	Don't mar/stain items	7.5	3.0	1.0	22.5	11%
	Easy to push or clamp on	7.3	1.0	1.2	8.8	4%
Work well over time	Last a long time	6.9	3.0	1.0	20.7	10%
	Resist weather damage	3.6	1.0	1.0	3.6	2%
Don't brake or tangle	Don't brake/ come apart	7.6	3.0	1.2	27.4	13%
	Don't tangle	2.4	1.0	1.2	2.9	1%

Figure 6-2 - Pre-planning chart, calculated by Multiplication of Scores Method [69]

The traditional method has two major drawbacks. On the one hand, the importance ratio between the different criteria is undefined. In the example above, the values of the competitive gap range from 1 to 9, the values of the sales point only from 1 to 1.5. A high value in the competitive gap criterion will impact the row weight much more than even the highest value in the sales point field.

On the other hand, the criteria are not independent from each other. Because of the multiplication, high values in the competitive gap and the sales point will interact and lead to a

[68] Yoder 1994, pp. 213 ff

[69] Day 1993, p.53

very high value in the row weight. This effect can be reduced by altering the criteria scales used by each criteria, but there is no way to judge when the composite rankings are correct.

6.3.2 Multiplication of Ratio Scores

Another method of customer want prioritisation is the development of ratio scales for each criteria which are multiplied to calculate the row weight for each customer requirement. When using this method, the criteria are used as filters to diminish the customer importance rating in an amount proportional to the amount by which the wants do not meet the criteria. Highly important criteria reduce importance ratings of customer requirements which do not fully the criteria by a larger amount than less important criteria.

An example is illustrated in Figure 6-3.

		Customer Importance	Comp. Gap (100%, 95%, 90%, 85%)	Reliability Issue (100%, 90%)	Sales P. (100%, 85%, 75%)	Abs. Weight	Rel. Weight
Work well on laundry	Grip things tightly	9.2	1.0	0.9	1.0	8.3	26%
	Don't mar/stain items	7.5	0.9	0.9	0.8	4.6	14%
	Easy to push or clamp on	7.3	0.9	0.9	0.9	4.7	15%
Work well over time	Last a long time	6.9	0.9	1.0	0.8	4.7	15%
	Resist weather damage	3.6	0.9	1.0	0.8	2.3	7%
Don't brake or tangle	Don't brake/ come apart	7.6	0.9	1.0	0.9	5.8	18%
	Don't tangle	2.4	0.8	0.9	0.9	1.5	5%

Figure 6-3 - Pre-planning chart, calculated by Multiplication of Ratios Method

Although the drawbacks of the Multiplication of Scores Method could be reduced, the method still suffers from the interaction of multiple criteria. The ratios by which each criteria reduces the composite importance ratings are difficult to determine. And finally, the composite requirements weights have no understandable definition.

6.3.3 Addition of Scores

Richard Zultner proposes a third method for calculating the requirements weights is to calculate the value by adding the scores from each column to arrive a combined score for each voice which describes the composite priority.[70] The column have different factors which represent their importance for the weight of the requirement. In Figure 6-4, the factors for the criteria are 50% for the customer rating, 30% for the sales point and each 10% for Competitive Gap and Reliability Issues.

[70] Zultner 1991

		Customer Importance	Comp. Gap (1,3,5,9)	Reliability (0,1)	Sales P. (1,1.2,1.5)	Abs. Weight	Rel. Weight
Column Weight		**50%**	**10%**	**10%**	**30%**	**100%**	**100%**
Work well on laundry	Grip things tightly	9.2	9.0	0.0	1.5	6.0	22%
	Don't mar/stain items	7.5	3.0	0.0	1.0	4.4	16%
	Easy to push or clamp on	7.3	1.0	0.0	1.2	4.1	15%
Work well over time	Last a long time	6.9	3.0	1.0	1.0	4.2	15%
	Resist weather damage	3.6	1.0	1.0	1.0	2.3	8%
Don't brake or tangle	Don't brake/ come apart	7.6	3.0	1.0	1.2	4.6	17%
	Don't tangle	2.4	1.0	0.0	1.2	1.7	6%

Figure 6-4 - Pre-planning chart, calculated by Addition of Scores Method

For instance, the row weight for "Grip things tightly" was calculated as:

$$0.5 * 9.2 + 0.1 * 9 + 0.1 * 0 + 0.3 * 1.5 = 6.0$$

This process is extremely flexible. Any number of criteria may be used without getting interaction problems. The weight of any column can be determined quickly and easily.

6.4 Computer Support for QFD-Charts

The generation of the pre-planning chart and any other QFD chart is an iterative process[71]. Many values will have to be changed during the QFD process, column weights or calculation methods can be modified. Especially when dealing with large amount of data and complicated calculation methods, the re-calculation of the charts is time-expensive and can be done more efficiently with computer assistance.

Due to the variety of different calculation methods and criteria that can be found in every single matrix of QFD projects, a flexible and powerful module is required that covers all different approaches. This universal module should provide both flexibility and performance in all parts of table definition items:

- Flexible number of criteria (fields)
- Flexible field types (values, formula, symbols, text)
- Flexible display of values (eg flexible number of decimals)
- Automatic chart calculation
- Flexible calculation method
- Intuitive formula syntax used for calculation
- Implementation of a variety of predefined functions

[71] Yoder p. 223

The implemented solution in the QFD-Manager covers all the above listed requirements. The pre-planning chart fields, their type and their calculation method are defined in the QFD-Manager's module *Field-Manager*. After the fields are defined, values for the fields can be entered in the *Data-Manager*. The final calculation of the formula values will be discussed in chapter 8.7.

6.5 Chart-Handling in the QFD-Manager

6.5.1 The Field-Manager

In the QFD-Manager, charts are represented by database tables. The space for the data is provided in fields, the actual information is carried in records (see chapter 2).

To add and define fields and store necessary information about the criteria they represent, the QFD-Manager has a built-in module which is called *Field-Manager*. The task of the Field-Manager is the definition of fields in database tables that are used as criteria in QFD charts. A screenshot of the Field-Manager is depicted in Figure 6-5.

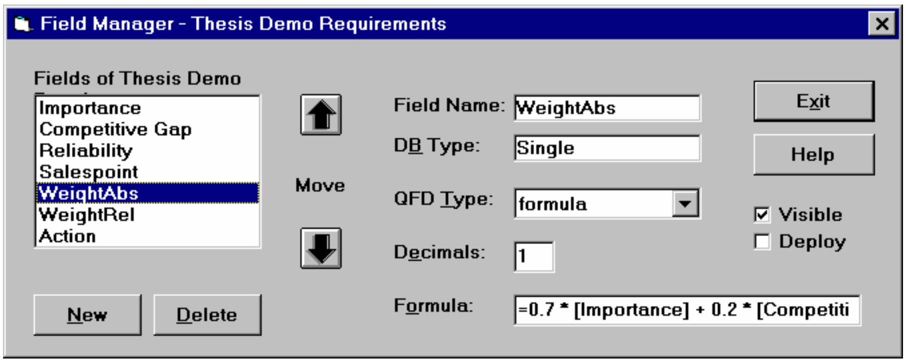


Figure 6-5 - The Field Manager

Not only the fields for a pre-planning chart can be defined in the Field-Manager. The module is a universal tool for generating all sort of database tables that are used in QFD-Matrices. For example the technical portion of the House of Quality (see Figure 6-1) or a Concept Selection Matrix (see chapter 9.3) can be designed with the Field-Manager.

To define a field, the system needs information about:

- the name of the field
- the type of the field (value, formula, text or symbol)
- the formula for the field (if field type is formula)
- the name of the database table which comprises the field's data
- field attributes

The name of the field is a short description of the purpose of the field. It can contain up to 64 characters.

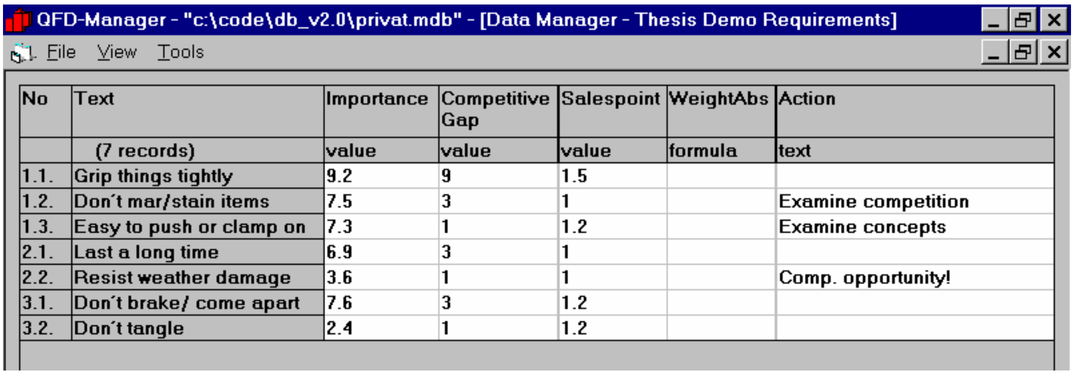
The field type is selected out of four different types: *value, formula, text* or *symbol*. If *value* is selected, the data for this field has to be entered manually in the Data-Manager or is provided by other parts of the system. For instance, the values of the importance field of a requirements table can be generated by the survey module. If the field type is *formula*, the values for this field are calculated automatically. To define the calculation method, a formula can be entered. A description of the formula syntax, the implemented functions and the formula interpretation algorythmus is given in chapter 8.7.

Finally, additional data for the field is stored in *field attributes*. The attributes give information whether the field will be displayed in print-outs of the matrix and whether the field will be deployed in later stages of the QFD-process (see chapter 8.9).

This set of information is stored in a separate system database table that is hidden for the user.

6.5.2 The Data-Manager

After the fields of a table are defined in the Field-Manager, data for the fields can be entered using the Data-Manager. A screenshot of the Data-Manager is illustrated in Figure 6-6.



No	Text	Importance	Competitive Gap	Salespoint	WeightAbs	Action
	(7 records)	value	value	value	formula	text
1.1.	Grip things tightly	9.2	9	1.5		
1.2.	Don't mar/stain items	7.5	3	1		Examine competition
1.3.	Easy to push or clamp on	7.3	1	1.2		Examine concepts
2.1.	Last a long time	6.9	3	1		
2.2.	Resist weather damage	3.6	1	1		Comp. opportunity!
3.1.	Don't brake/ come apart	7.6	3	1.2		
3.2.	Don't tangle	2.4	1	1.2		

Figure 6-6 - The Data-Manager

Also included in the Data-Manager Module is an extraction filter. The displayed records and fields of the database table can be selected by various criteria. The data of any database table that is used in QFD matrices can be entered, modified or deleted.

Data can only be entered in fields of the type *value*, *text* and *symbol*. Direct access to the data of *formula* typed fields is blocked, because this data will be calculated automatically when calling the *Calculate* function (see chapter 8.7).

7 The Technical Information Table

In the preceding chapters, the completion of the House of Qualities' horizontal portion, often called pre-planning chart or customer information table, was discussed. Customer voices were sorted out and transformed to customer requirements, they were grouped on different levels of hierarchy, and a customer ranking and competitive evaluation was performed. The survey data provided further information for the requirements, and additional data could be placed in the fields of the pre-planning chart.

This chapter discusses the vertical or technical portion of the House of Quality (see Figure 7-1). This part of the matrix is called the *Technical Table*.

Before starting to work on the technical table, the list of customer requirements has be to completed, but it is not necessary to have accomplished to customer ranking, customer competitive assessment and issues of further pre-planning information. If time is a critical factor for the QFD team, these tasks can be worked on simultanously with the technical product characteristics. The technical translation of customer requirements can begin while another part of the team works on the qualitative survey to perform customer importance rankings and competitive evaluations. There is no specified order in which information is collected for the House of Quality[72]. No particular order of completion should be forced, because teams have to decide in each case which way suits best to solve their problem.

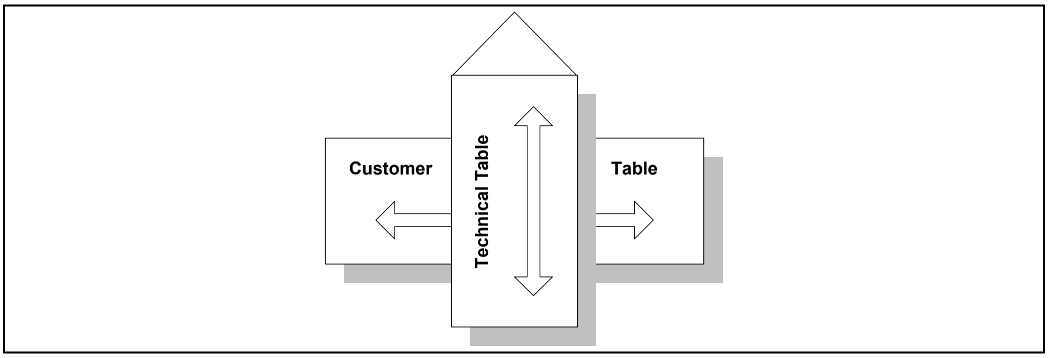


Figure 7-1 - The Main Tables of the House of Quality

7.1 Translating Requirements into Characterisitics

The first step of beginning the technical table is the translation of the customer requirements into *Technical Characteristics*. In current literature, the term varies. Commonly used expressions are *Design Characteristics*, *Technical Requirements*, *Technical Equivalents*, *Product Characteristics*, *Quality Characteristics* and *Performance Measures*. In this thesis and in the QFD-Manager, the simple term *Characteristics* is used to separate the technical items from the customer requirements, which are simply referred to as *Requirements*.

[72] Day 1993, p. 63

The requirements must be translated into the type of language the company uses to describe its products for design, processing and manufacture. The requirement information is too fuzzy for the design process because the language of the customer is not very specific. Words such as *quickly, quietly, efficiently* must be changed to the *engineer's language*, such as *time to complete* or *frequency*. Measurable performance indicators are necesssary to evaluate alternative designs and to predict the satisfaction of the customer.

Terninko describes technical characteristics as "a technical measurement evaluating the product's performance (of a customer requirement)"[73]. King defines: "The items that a producer controls to assure that they meet customer demands. The state how the producer meets demands. [..]"[74]

Each requirement is translated into one or more technical characteristic. A characteristic should be[75]:

1. something that should be worked on to satisfy a voice,
2. measurable,
3. global in nature, not implying a certain design intent or solution.

The intent of the technical portion of the House of Quality is not to imply design solutions. This task is performed later in the concept generation phase or subsystem or parts planning stage. The purpose of the technical portion, instead, is to provide a series of technical characteristics that specify a "generic design that responds to the customer [...] (requirements)"[76].

The translation process is one of the hardest part during the QFD process, because people have a natural desire to think in solutions. This can be illustrated best by an example[77]. When customer's voice states "Want to stop straight, especially in emergency situations and on wet or slippery roads", people tend to think about ways to accomplish this want. The team might think about improving tires or installing an anti-blocking braking system, to provide more consistent straight-stopping. Instead, the search has to focused on ways of measuring the product's characteristic that satisfies the customer requirement. A typical *measurable response* to the requirement might be *variation from straight topping line during stopping.* The less variation, the higher the customer's satisfaction. The best concept for accomplishing this can be chosen later in subsequent matrices of the QFD process.

In Figure 7-2, an example for translating customer requirements into technical characteristics is given for a coffee cup.

---

[73] Terninko 1995, p. 86

[74] King 1989, p.2.3

[75] Day 1993, p.64

[76] Day 1993, p. 64

[77] Day 1993, p.64

Requirement	Technical Characteristic
Cup stays cool	Temperature at hand
Coffee stays hot	Fluid temperature loss over time
Won't spill/tip	Tip force at top Fluid loss over vertical impact Fluid loss horizontal impact
Resists squeeze	Indent/force relation Force/set relation
Doesn't leak	Porosity

Figure 7-2 - Translating Requirements into Characteristics

Technical characteristics should be determined with care. Generally, one or two characteristics are necessary to satisfy a requirement. It is important not to produce too many characteristics, because the level of detail will be too high and the complexity of the House of Quality increases measurably. The number of characteristics determines the number of columns in the matrix. This number predicts the complexity if work to be done in later stages. Not only more relationships to the customer requirements have to be determined, but also the difficult co-relationships between technical characteristics will increase quadratically. The expenses for the tests to develop the competitive technical assessment data will increase as well. Day suggests to limit the ratio of technical characteristics to customer requirements between 1 and 1.5[78].

Every characteristic has to be *measurable product parameter*. As a check for *measurablity* of the characteristic, the measurement unit should be determined directly when creating a new characteristic. This methodology ensures, that non-measurable characteristics are avoided right from the start.

Once all characteristics are determined, the can be arranged into natural groups using the affinity process. This task is especially advisable when dealing with larger matrices. The methodology is similar to the requirements grouping process described in chapter 4. In the case of technical characteristics, the grouping procedure is often done by arranging the items by *functional responsibility*.[79]

7.2 Improvement Directions

For every technical characteristic exists a direction that customer prefer. It is helpful for the QFD team to define the direction of increasing customer satisfaction for the characteristics. The information is most helpful when examining the interactions of the technical characteristics. These co-relationships are strongly influenced by the direction of increasing customer satisfaction.

The direction of improvement is illustrated with the help of symbols. Figure 7-3 shows symbols that are commonly used by many companies. The arrow pointing up is used to indicate that the customer would prefer the value of this product characteristic to be larger, bigger, heavier,

[78] Day 1993, p. 70

[79] Day 1993, p. 70

more, or in general, higher. By contrast, the arrow pointing down indicates slower, smaller, lighter, shorter, or in general, lower. On the other hand, there are cases when the customer is prefered best by meeting a certain target value. An example for a target could be the distance of the steering wheel of a car from the shoulders of the driver. The customer's satisfaction will be maximized, when the distance is "in reach", that means, not too far, not too near.

There are also some cases in which the customer has a direction of improvement, but only up to a certain point. If the values reaches this point, any further improvement on the characteristic has no effect. For instance, the temperature on the outside of a coffee cup filled with hot coffee should have a certain maximum value to satisfy the customer requirement "Cup stays cool". If there is any variation from this target value, the temperature should be on the lower side side of the target, that means, cooler. On the other hand, a cooler cup will not lead to more customer satisfaction. As long as the customer doesn't burn his hand, he is still satisfied. In the technical table, a circle with an arrow pointing down would be used.[80]

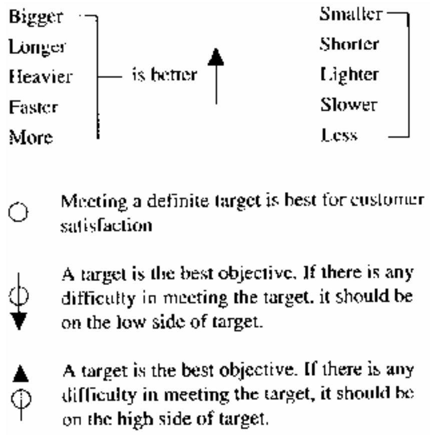


Figure 7-3 - Symbols for Improvement Directions of Characteristics[81]

Kommentar: "Direction Symbols From Day2.bmp"

The selection of the direction for the characteristics can be made during the translation process. It can be recorded directly with the characteristic and its units of measurement.

After the directions are selected, a base of the technical table is complete. We have now detailed information of how to translate the voice of the customer into the voice of the engineer. The characteristics form a quantitativ measurement system for the quality of our product. Customer satisfaction is made predictable.

7.3 Technical Competitive Data and Target Values

After the technical characteristics have been established as a measurement system for the quality of product, we apply the system for setting target values for the new product. By means

[80] An interesting connection can be designed between the directions of design characteristics and the Taguchi Loss Function. For a detailed description see Terninko 1995, p. 107ff

[81] Day 1993, p.73

of these target values, the total quality of our product is defined. This is the last step of making customer satisfaction both predictable and plannable.

Before setting up target values, it is essential to perform a technical competitive analysis. This can be performed within the company by testing the characteristic values of our own and the competititors' products. As soon as the technical characteristics have been established, the team should begin to arrange the testing. One of the major influences affecting test time is the number of competitor's products that will be evaluated[82]. The total test time and expense will increase in proportion of to the number of tests scheduled. Thus, the team should select carefully which products to assess. A number of tests should be selected that represents a balance between total test time and the need for information. Generally, the customer competitive analysis can give valuable hints for the selection process. At least one of the top-class products of customer satisfaction, and one of the lower-class products should be tested to get a balanced market overview.

Figure 7-4 shows sample test results from a technical competitive assessment.

Customer Requirement	**Technical Characteristic**	**Competitive Assessment**			**Units**
		A	**B**	**C**	
Resists squeeze	Indent/force relationship	0.17	0.23	0.36	inch per lb
Coffee stays hot	Temp. loss over time	4.9	4.2	3.9	degrees F/min
Cup stays cool	Temp. at hand	150	130	148	degrees F
Easy to operate	Operating effort	8.6	6.8	4.6	ounces

Figure 7-4 - Results from technical competitive assessment[83]

When the technical competitive tests are complete, the discussion about target values can begin. Usually it is best, to assess the relationships between customer requirements and characteristics, before starting the target discussion (see chapter 8.5). The setting of targets is a difficult task, which is influenced by many parameters. The more parameters are known during discussion, the better the quality of the target values will be. Main influence factors for the targets are:

- the relationships values for the corresponding customer requirements,
- customer importance rankings of the requirements,
- customer competitive assessment data of the requirement,
- additional data from the preplanning chart, eg the number of complaints.

Beside this data, it can be essential to gather additional data about certain technical characteristics before setting targets. Eg in our coffee-cup-example, it could be necessary be find out what the maximum temperature at hand is, that customer still consider to be comfortable.

Further information and examples about competitive analysis and target setting can be found eg in Day 1993, p. 75ff.

---

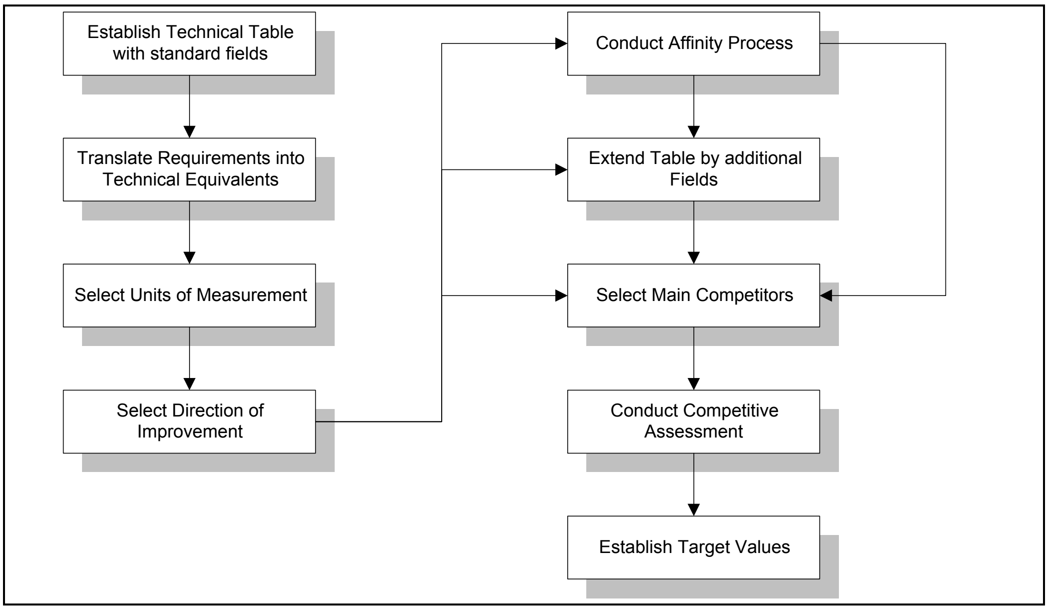
[82] Day 1993, p.76

[83] Day 1993, p.77

The technical table can be enlarged by additional fields. Day suggests to include fields as *Organizational Concerns*, *Field Experience*, and *Regulatory Technical Company Issues*. The Organizational Concerns comprise indicators about problems that will occur when implementing a certain technical characteristic to the product because of the organizational structure of the company. Field experience indicates areas of product development, that the company is already experienced in. These characteristics will be easier to develop for the company. Regulatory Technical Company Issues comprise constrains that the company has have to fulfill. These could be for example environmental laws or other issues that have to be satisfied. If not, the customer would merely buy a single unit of the product.

7.4 Summary of Generating Technical Characteristics

Figure 7-5 illustrates the development steps of the Technical Table. The first four steps including the translation process, the selecting of measurement units and the direction are compulsory, because they are essential for the House of Quality. The following steps can be done if the team considers them to be important.



Kommentar: tech.vsd

Figure 7-5 - Development of Technical Table

The Affinity Process should be conducted if the technical characteristics exceed certain number. Sorting them in different levels and groups can increase overview and consistency of data. It is easier to check if important characteristics are missing or if similar items could be merged to a single one.

If additional data shall be included, the team can add fields to the chart that provide room for the information. After the competitors products are selected, the technical competitive analysis can begin. When all data is provided, the team discusses the final target values.

7.5 Computer Assistance for Technical Characteristics

When establishing computer support for the development process of the technical characteristics, we have to consider that the translation process is rather a creative process than a "technical procedure”. A computer can’t substitute the human creativity that is needed to perform the task.

On the other hand, software can support the translation process by offering a screen mask that displays a customer requirement and a textbox for entering the technical characteristic (see Figure 7-7). In the implementation of the QFD-Manager, the term *Technical Equivalent* is used to stress the direct translation of a customer requirement into a technical characteristic.

After QFD-Manager’s *Characteristics Module* is started, the user selects the source database table containing the requirements that are to be translated. After that, a database table for the technical information can be selected or created. This table is called a *characteristics table*, and contains the technical characteristics and all related information.

The basic structure of the table is the same as of the *requirement table* (see chapter 3.7.2). Using this basic structure, the table can be used in all other parts of the QFD-Manager (see chapter 4.4.3). When creating a new characteristics table, the table definition is enlarged by some fields that carry technical information as depicted in Figure 7-6. All field definitions can be removed, changed or enlarged using the Field-Manager (see chapter 6.5). By providing this flexibility, the QFD-Managercan be adapted to every QFD project.

ID	Text	Comment	Level	Parent	Position
long int.	text	text	integer	long int.	integer

Direction	Measurement	Company Now	Target	Difficulty	Absolute Weight	Relative Weight
integer	text	single	single	single	single	single

Figure 7-6 - Structure of Characteristics Table

The characteristic is contained in the field *Text*. The next fields contain the grouping information that can be filled by the Affinity Module later. *Direction* contains an integer value. This value refers to one of the five possible optimization directions (see chapter 7.2). The field *Measurement* contains the units of measurement in text form. Using this flexible field type, every possible unit can be used (eg cm, inch, km/h, frequency, etc...). The field *Company Now* comprises the test values for the own product which is produced by the test series. Usually, the table will be enlarged by other fields containg the values of the competitors’ products. These fields could be named *Competitor A* or *Product XY*, and can be added using the Field-Manager, which was introduced in chapter 6.5. The field *Difficulty* can contain additional data as organization difficulty. If not needed, the field can easily be removed again with the Field-Manager. The fields *Absolute Weight* and *Relative Weight* contain the calculated importances of the characteristics. They are defined as automatically calculated fields. The detailed description of the calculating process is given in chapter 8.7.

When the requirements and characteristics tables are selected, the main screen appears (see Figure 7-7). The translation process is conducted by repetitiously asking the question *"What are the Technical Equivalents of this Requirement?”*. The requirement is displayed on the screen and the user can enter the technical equivalents.

To each equivalent, the units of measurement are defined directly after entering. That is not only the easiest way for the user, it also ensures that a correct equivalent has been found. If a measurement unit can't be entered, the item can't be a technical characteristic, because it can't be quantified. For that reason, the units of measurement are compulsory. The system will not save a characteristic to the database without it.

A direction of improvement is compulsory as well, and is selected after the units of measurement. The user does not have to deal with the internal code values in the database table, he can select the direction easily from a listbox. All five directions described in chapter 7.2 are available. The list of equivalents is displayed on the screen. The user can edit or delete items at any time.

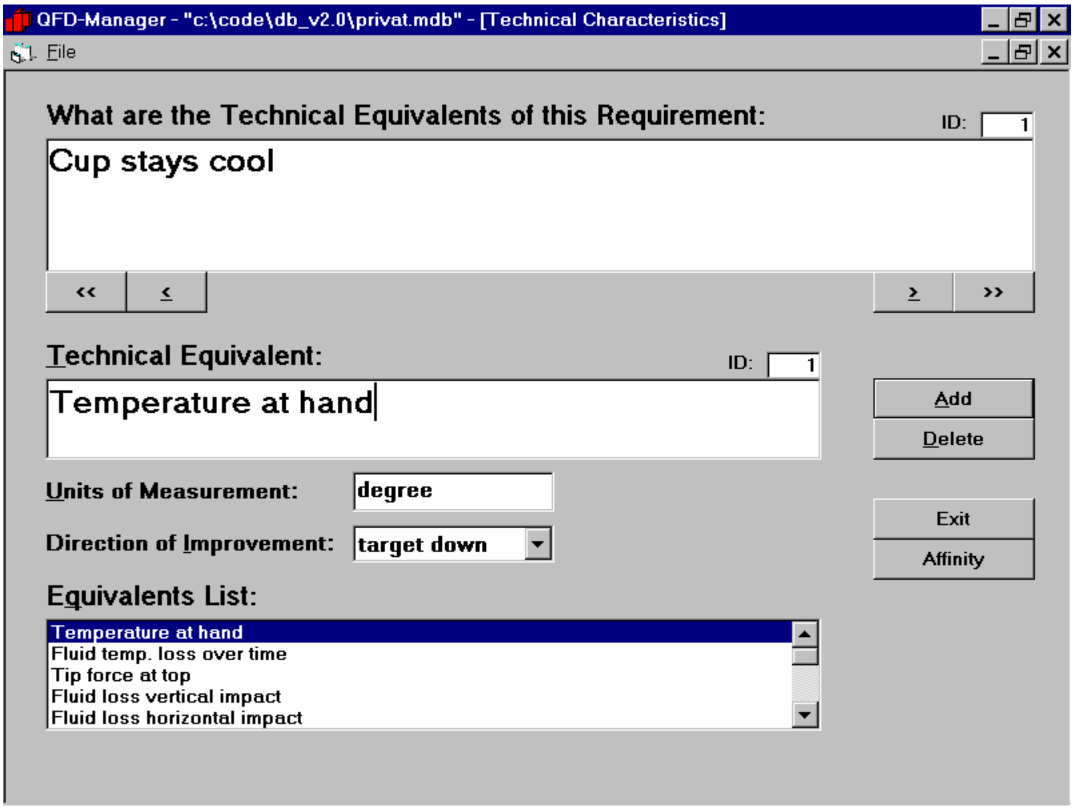


Figure 7-7 - The Characteristics Module

After all customer requirements are translated and the list of equivalents is completed, the user can branch to the Affinity Module, where the characteristics can be placed into associated groups.

Usually, the Technical Table has to be adapted to the particular QFD project. This can be done by branching to the Field-Manager. Fields for organizational concerns, field experience, etc. can be defined here. All data can be entered using the Data-Manager (see chapter 6.5.2).

The next step of developing the technical table is to select the competitors products that shall be included to the technical experiments. This task is usually a management or at least a team decision, that can't be supported by software. The team has to select carefully a number of products the represent best the different products currently available on market.

After the test data is established, fields for the competitors data are created with the Field-Manager. The test data can be entered using the Data-Manager. No further software support is necessary for this task.

Also the definition of target values can't be supported by software. The decisions made in this part are some of the most essential parts of the whole QFD process. The new product is defined quantitatively. The only support given by software can be a print-out of the QFD-Matrix that provides necessary data for the decision making process (see chapter 8.8).

8 Matrixhandling

8.1 Charts and Matrices in QFD

As described in chapter 6.1, the basic components of matrices are charts. The most popular example for a matrix in QFD is the House of Quality, which is formed by the customer table and technical table.

The purpose of QFD matrices is always similar. The matrix describes relationships between the items of the two tables which are displayed in the intersection of both charts as symbols or values. Further, matrices provide special means of data transfer from one table to another by using the relationship values and combining them using certain calculation methods.

Because the House of Quality occurs in almost every QFD project, the HoQ will be taken as an example to explain matrix calculation procedures. In other QFD matrices, the methods are almost identical. An exception is the Concept Selection Matrix, which will be discussed in chapter 9.3.

The main purpose of using matrices in QFD is to identify crucial items of a chart by relating them to the items of another chart. For instance, a major purpose of the House of Quality is to identify the most important design characteristics to fulfill certain customer requirements.

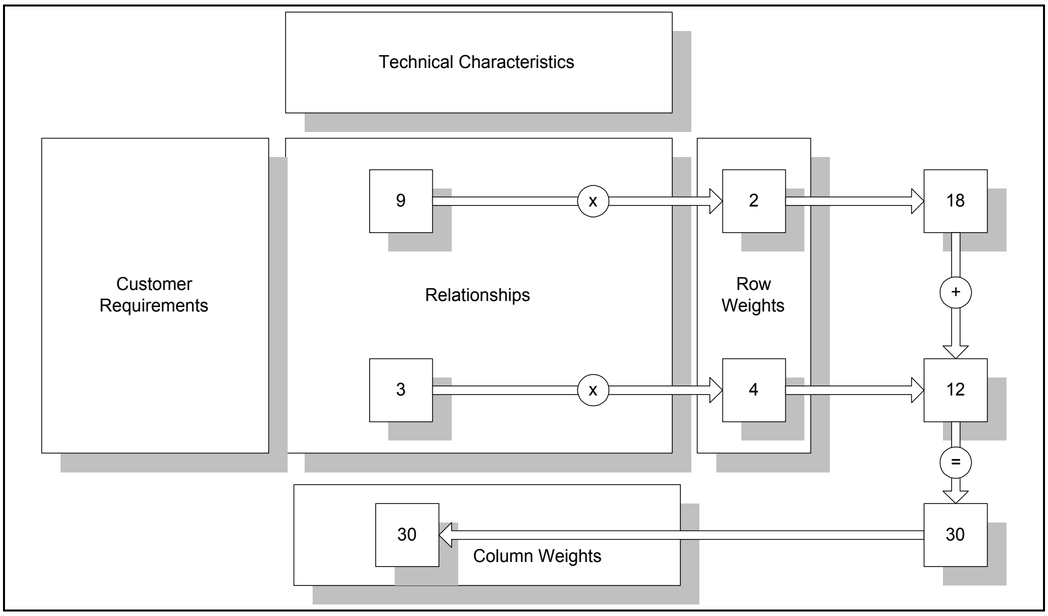


Figure 8-1 - Systematic of Matrix Calculation

Kommentar: Matrix Systematic.vsd

The importances of the design characteristics (*column weight*) are calculated by evaluating their relationships to the customer requirements (see Figure 8-1). Design characteristics that can help to fulfill customer requirements, have a strong relationship value (eg 9). If they are less strong related, they have a smaller value (eg 3 or 1). If no relationship exists, the relationship value is zero. In QFD matrices, the strength of relationships is usually illustrated by symbols.

To calculate the column weight, the relationship values are not just simply added, they are first weighted by multiplying them with the importance values or row weights respectively of the customer requirements. After that, the row products are summed up as importance of design characteristic. Because the values express the importance of a column, they are also called *column weights*.

In the example in Figure 8-1, a certain design characteristic helps to fulfill two customer requirements. The relationships values (9 and 3) are each multiplied with the importances of the respective requirements, which are 2 and 4. The final result (18+12=30) represents the column weight.

After the procedure is performed for all design characteristics, the calculated columns weights can be normalised. The normalised weights represent the importance of the design characteristics in percent.

An exception to the presented way of calculating QFD matrices is the concept selection matrix. Concept selection and the related matrix calculation methods will be discussed in chapter 9.3.

8.2 Necessity for Computer Support

Now we know enough about QFD matrices and charts to sort out possibilities for computer assistance.

The necessity for computer assistance for QFD matrices is obvious. QFD was introduced in the 1960s as a *paperbased* Method, because computer technology was in very early stages. The QFD matrices were centrally modified, updated, and copied to the members of the QFD teams. All work was done manually[84].

Lyman states, that "one of the biggest deterrents to using QFD is the lack of tools”[85]. Important drawbacks of the paperbased method are:

- The manual work to modify, update and reproduce the matrices is onerous and time-expensive.
- Even when dealing with smaller QFD matrices, the re-calculation task is very expensive. To calculate a matrix with 25 customer requirements and 40 design characteristics, up to 2000 operations can be necessary.
- Mistakes occur regularly when calculating matrices manually.
- The paperbased method is not feasible for larger sets of data, because the data doesn’t fit on paper sheets.
- Correcting data on paper sheet is time-expensive and people tend to loose overview when the sheet gets dirty.
- Automatic Search- and Sort-Functions can’t be applied.
- Standard paper sheet for QFD matrices are inflexible when adapting the project to special tasks and purposes.

In the QFD-Manager, we try to overcome these drawbacks by offering a tool the covers all tasks related to handle QFD matrices.

---

[84] Fowler 1991, p.178

[85] Lyman 1990, p.312, in: Westphal 1996, p. 94

8.3 The Matrix Module

In the QFD-Manager, a QFD matrix is defined as a combination of at least three database tables. Two of them carry the information of the horizontal and vertical chart of the matrix, the third contains the relationship information between the chart items. Additionally, a fourth database table can be defined carrying co-relationships between technical characteristics, although this database table is not compulsory.

Matrices are defined in the QFD-Manager's *Matrix Module*. A screenshot is displayed in Figure 8-2.

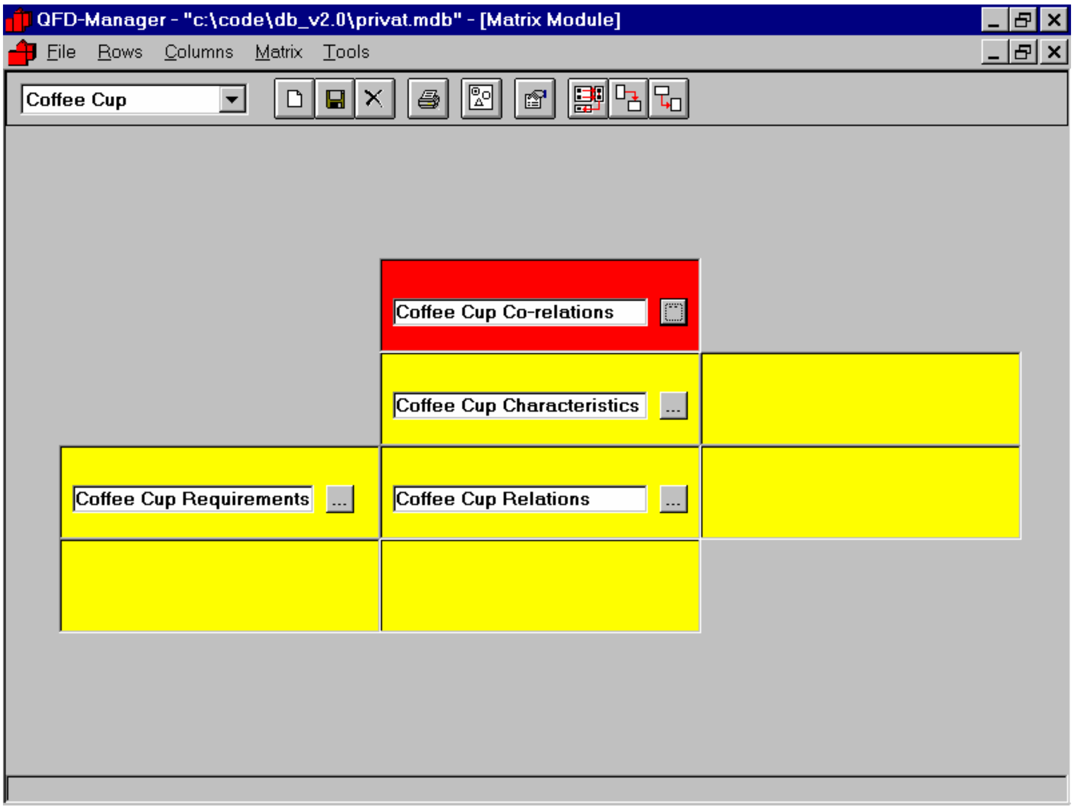


Figure 8-2 - The Matrix Module

The definition of a matrix comprises the following information:

- A unique identification number for the matrix (ID)
- Name of the matrix
- A comment which can contain for example the purpose of the matrix or other related information
- The matrix type, indicating whether the matrix is used as a concept selection matrix
- Attributes, containing additional information about the matrix

- Names of the database tables operating as source tables for the matrix

The matrix definitions of a QFD project are stored in a system database table called *Matrix Definition Table*. The table is hidden for the user. Its structure is displayed in Figure 8-3.

ID	Matrix name	Comment	Type	Attributes	Row table	Column table	Relations. table	Co-rel. table
long int.	text(64)	text(255)	long int.	long int.	text(64)	text(64)	text(64)	text(64)

Figure 8-3 - Matrix Definition Table

8.4 Functions of the Matrix Module

The task of the Matrix Module is to manage matrix definitions, to handle the matrix content, and to perform matrix related QFD procedures. To perform this task, the following functions are included in the Matrix Module:

- Handle matrix definitions

  To create a new matrix, the user select the respective menu item and enters a matrix name. After that, database tables being used as source charts in the matrix are selected. Not all parts of the matrix have to be filled with tables. It is for example possible not to specify the vertical chart and the relationships matrix. Then the matrix consists only of one chart, eg a pre-planning chart, that many matrix functions can still be used on.

  After a matrix is defined, the matrix definition is stored into the Matrix Definition Table, from which it can be recalled later.

- Manage matrix content

  More interessting than the pure matrix definition is what you can acutally do with your matrix and its content. The Matrix Module's features all functions to specify and modify the structure and data of QFD matrices.

  To define fields in the horizontal and vertical part of the matrix, a link to the Field-Manager (see chapter 6.5) is provided. Fields can be added, removed or modified, formulas for chart and matrix calculation can be specified. To handle the underlying data of the fields, branches to the Data-Manager can be selected, in which the field values can be displayed and modified (see chapter 6.5.2).

  The specification of the relationships between the items of the charts of the most important part of the Matrix Module. To enter relationships, the Relationship-Manager can be used (see chapter 8.5) featuring symbols and drag-and-drop functions. The same method applies for co-relationships.

- Calculate matrix

  After all matrix formulas are defined in the Field-Manager, the automatic matrix calculation can be performed. For a closer description of the formula syntax, the provided standard functions and the calculation procedure please refer to chapter 8.7.

- Output

  To provide a good overview of the completed QFD matrix, the matrix can be displayed entirely or partly on the screen. The user can select different zoom factors. Also, the matrix can be printed out on every MS-Windows compatible Printer. The basic considerations to realize this part are described in chapter 8.8.

- Provide data interface

  One of the most significant drawbacks of traditional QFD-Software is its incompatability between each other. To overcome this disadvantage, the proposed QFD Data Exchange Standard (QFD/DES) is used as an interface to import and export QFD matrices[86]. The QFD/DES is realised eg in the QFD/Capture Software of ITI[87].

- Deploy matrix data

  When the content of a matrix is complete, the matrix can be deployed. By deploying a QFD matrix, its data can be used in other parts of the QFD process. The Macabe-Modell uses a simple four-matrix deployment, more sophisticated QFD methodologies like the Akao-Modell, the King-Modell or Comprehensive QFD are performed by many other forms of deployment. The QFD-Manager provides a flexible deployment functionality which is described in chapter 8.9.

8.5 The Relationship-Manager

The Relationship-Manager is the core of the Matrix Module. It assigns relationships of the row and columns items of the matrix.

Relationship values can have different meanings, depending on the matrix definition and purpose. In the House of Quality, where *Customer Requirements* are related to *Technical Characteristics*, the relation value could mean *"degree of how the product characteristic is capable to fulfill a customer need"*. When *Functions* are related to *Characteristics*, the value could express the *"degree of how the product characteristics provides a certain product function"*.

To illustrate the strength of the relationships between QFD matrix items, generally symbols are used. The underlying value of the symbols may vary in different QFD projects; to keep the QFD-Manager flexible for every QFD application, the symbol values can be modified by the user. Standard values are adapted from the wide-spread Japanese QFD symbol set[88], in which a triangle represents the value "1", a bordered circle "3", and a filled circle "9". If relationship values shall be entered that don't fit into the symbol set, they can be specified by entering them as numerical values. This option guarantees a maximum of flexibility.

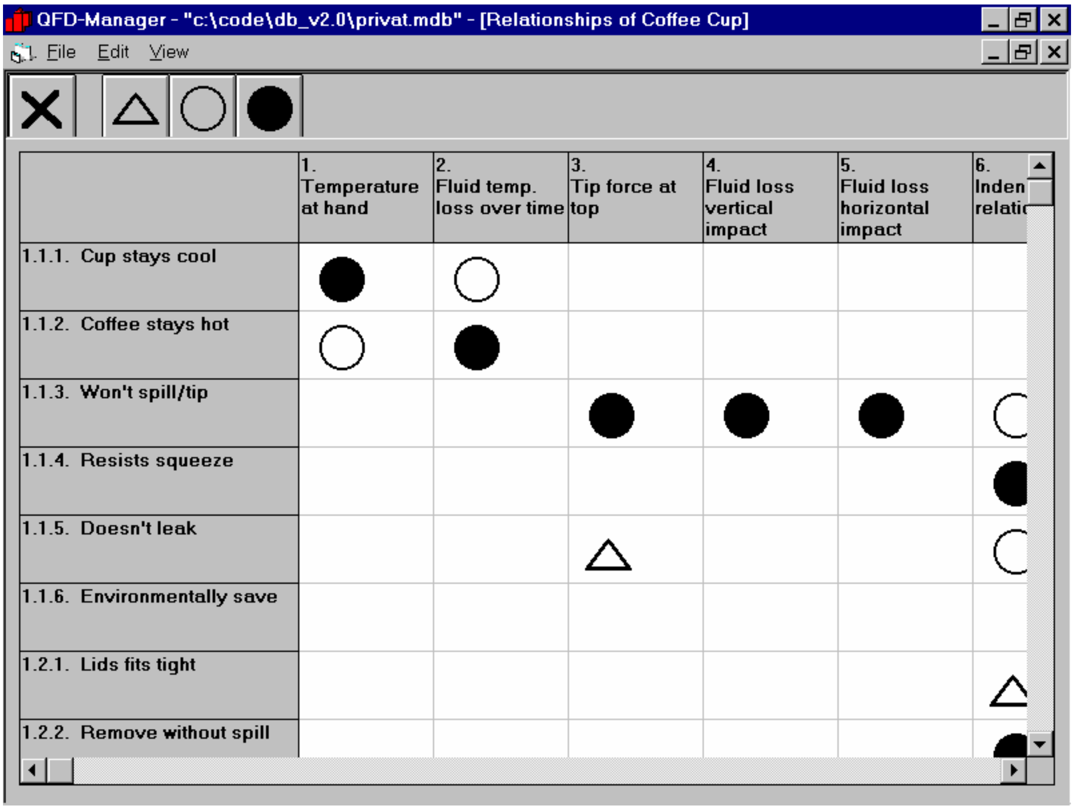
In the Relationship-Manager, the symbol set is displayed on a toolbar above the matrix. There are different sets of symbols available, a set for matrix main relationships, a set for co-relationships, and a set for concept selection matrices. In Figure 8-4, a screenshot of the Relationship-Manager is illustrated, displaying the relationships for a House of Quality matrix.

---

[86] Hales 1991

[87] International TechneGroup Inc. 1991, pp. H4-H19

[88] Guinta 1993, p. 53. Guinta asserts, that the symbols are derived from the popular Japanese horse racing sport.



Kommentar: relaman.bmp

Figure 8-4 - The Relationship-Manager

The symbols can be assigned by using the drag and drop functionality of MS-Windows. That means, that a symbol is chosen from the toolbar by clicking the mouse button, holding it and dragging the symbol to the field that represents the relationship of two items. When the mouse button is released, the symbol appeares in the relationship field.

As in the Data-Manager, many options for extracting and displaying items of the two matrix charts can be selected. After the chart is completed, the values are saved into the relationship table that was specified in the Matrix Module. The structure of the relationship table is discussed in the following chapter.

8.6 Structure of the Relationship Table

The relationship values specified in the Relationship-Manager have to be stored into the database to preserve them for later QFD-Manager sessions and to make them available for matrix calculation.

Considering how to store the relationship information in databases, we have to remind the relationship structures of chapter 2.5 about databases. Obviously, the kind of relationship between the two charts of the matrix is a many-to-many relationship. The minimum cardinality of both tables is zero, because an item does not necessarily have to have a relationship. The maximum cardinality is defined as the number of relationships an element of a table can participate in, which is the number of elements in the opposite table.

To store data assigned to many-to-many relationships, *crosstables* are used. A crosstable contains at least two pointers, each pointing to an element of another database table. To each of these combinations, data can be assigned. In our case, this is the strength of the relationship as a numerical value, and a comment. Relationships that have a zero-value are redundant for displaying and calculating the matrix and do not appear in the crosstable. An example for the QFD-Manager Relationship Crosstable is illustrated in Figure 8-5.

ID1	Text
1	Cup stays cool
2	Coffee stays hot
3	Won't spill/tip
4	Resists squeeze
5	Doesn't leak
6	Environmentally safe
7	Lids fits tight

ID1	ID2	Relation
1	1	9
1	2	3
2	1	3
2	2	9
3	3	9
3	4	9
5	3	1

ID2	Text
1	Temperature at hand
2	Fluid temp. loss/time
3	Tip force at top
4	Fluid loss vertical impact
5	Fluid loss horiz. impact
6	Indent/force relation
7	Puncture resistance

Figure 8-5 - Relationship Crosstable

The proposed method allows to store the relationship data in an easy and flexible way. Further, it allows other applications to use or provide data in the QFD-Manager format without serious compatability problems.

8.7 Field and Matrix Calculation

8.7.1 Field Formulas

After the relationships are defined, all data is provided to calculate the matrix. As described in chapter 6.1, there are many different ways to define matrix fields and methods to calculate weights and other fields.

To define a field that is calculated automatically as an arithmetic combination of other fields within the QFD-Manager, formulas are used. The formula of a field is specified in the Field-Manager (see chapter 6.5) together with the field definition.

Using formulas is the most powerful and flexible way to perform the matrix calculation. No other way can provide such a great variety of functions and algorythms, and cover almost all possible ways of calculating matrices.

Every formula starts with an equal sign ("="). The syntax of a formula can comprise the elements displayed in Figure 8-6 in descending priorisation order. Priorisation order means, which operations are performed first. For instance, the multiplication operation has a greater affinity than the addition operation, and is conducted first. Generally, the syntax of formulas in the QFD-Manager follow the general rules of basic mathematic books.

Element	Meaning
[fieldname]	Representation of a fieldvalue or reference to a field. Always embraced with []
Constant	Constant numerical values, eg 3.14
()	Priorisation Braces, eg (2 + 3) * 4
Function()	Functions, eg max(), matrix(), or pugh()
*,/	Multiplication and Division Operator
+,-	Addition and Subtraction Operator

Figure 8-6 - Formula Elements

Figure 8-7 illustrates the provided function set. Functions can have arguments which are embraced behind the function name. Arguments can be constant values or fieldname references.

Functions	Meaning
min(value1, value2, ...)	Returns the minimum value of value1, value2, ...
max(value1, value2, ...)	Returns the maximum value of value1, value2, ...
fix(value)	Cuts all decimals of the argument
abs(value)	Converts negative arguments into positives. Positive values remain unattached.
bool(value)	Returns 0 if argument is equal to 0, else 1
sign(value)	Returns 1, if argument positiv, 0 if zero, -1 if negative
rnd()	Provides a random value between 0 (inclusive) and 1 (exclusive)
sin(value)	Returns the sinus value of the argument. Operator in degree
sum([fieldname])	Returns the sum column or row named fieldname
avg([fieldname])	Returns the average column or row value named fieldname
norm([fieldname])	Based on fieldname, a normalised value is calculated. After the calculation, the sum of all normalised values is 1.
matrix([fieldname])	Weights the relationship values of the relationships matrix with fieldname and calculates its sum row- or columnwise.
pugh()	Calculates the sum of all relationship values in the respective row or column. Used for concept selection matrices.
pughpos()	Calculates the sum of all positive relationship values in the respective row or column. Used for concept selection matrices.
pughneg()	Calculates the sum of all negative relationship values in the respective row or column. Used for concept selection matrices.

Figure 8-7 - Function Set of QFD-Manager

8.7.2 Formula Syntax

The syntax of QFD-Manager formulas can be condensed in the following ten rules:

1. The result of a formula is a numerical value.
2. Formulas can be constructed by providing one or combinating more than one values.
3. Values are combined with operators (+,-,*,/).
4. Multiplication and Division is conducted before Addition and Subtraction. If the priority of operators is equal, the calculation is performed sequential from left to right.
5. "(" and ")" Braces override the priority of operators.
6. Values can be represented by numerical values (eg 3.14), field references ([Salespoint]), or functions.
7. Numerical Values can contain a decimal point. Leading Zeros can be omitted (eg 0.7=.7).
8. Field References are embraced by "[" and "]" braces. Fieldnames are not case sensitive, that means that "[Salespoint]" equals "[saLESpoINT]".
9. Functions are constructed by the function name, opening "(" braces, arguments divided by commas, and closing ")" braces.
10. Function Arguments can be values or field references. The number and type of arguments depends on the function definition (see Figure 8-7).

The syntax is demonstrated best by giving some examples that actually could be relevant for QFD applications. The examples below occur in QFD matrices like the House of Quality regularly.

The following formula could be used to calculate a competitive gap for [My Product], compared to the performance of [Product 1], [Product 2], and [Product 3]:

[Competitive Gap] = max([Product 1],[Product 2],[Product 3]) - [My Product]

The functions would produce negative values, if there are combination if which [My Product] performs better than any of the competitors products. To exclude these cases, the formula could be further embraced with maximum function containing zero as the second argument. A zero is always greater than negative values, so negative values are excluded:

[Competitive Gap] = max(max([Product 1],[Product 2],[Product 3]) - [My Product] , 0)

Once the Competitive Gap is evaluated, the field could be used to calculate the absolute requirements weight, including the fields [Importance] and [Salespoint]:

[Abs Row Weight] = 0.7 * [Importance] + 0.2 * [Salespoint] + 0.1 * [Competitive Gap]

This value could be normalised to 100% in the field [Rel Row Weight]. The values of this field would represent the relative weight of each customer requirements.

[Rel Row Weight] = norm([Abs Row Weight]) * 100

For the technical requirements, the matrix() function would apply. The matrix() function performs the matrix calculation that is described in Figure 8-1. All relationship elementsof a column are weighted with the values a column of the horizontal chart. These values are added and represent the technical importance. In the QFD-Manager, the field [abs technical weight] could be defined as:

[Abs Technical Weight] = matrix([Relative Weight])

It is important to mention, that the argument [Relative Weight] of the matrix() function is not a reference to field in the same part, the vertical portion of the matrix, but of the opposite part, the horizontal portion of the matrix. The matrix() function can also be used on the horizontal part of the matrix. Then the calculation is performed analogous with a field of the vertical portion weighting the relationship values of a row.

For the relative technical weight could apply:

[Rel Technical Weight] = norm([Rel Technical Weight]) * 100

The same result could obviously be gained by:

[Rel Technical Weight] = norm(matrix([Relative Weight]) * 100

(example of House of Quality, with all values)

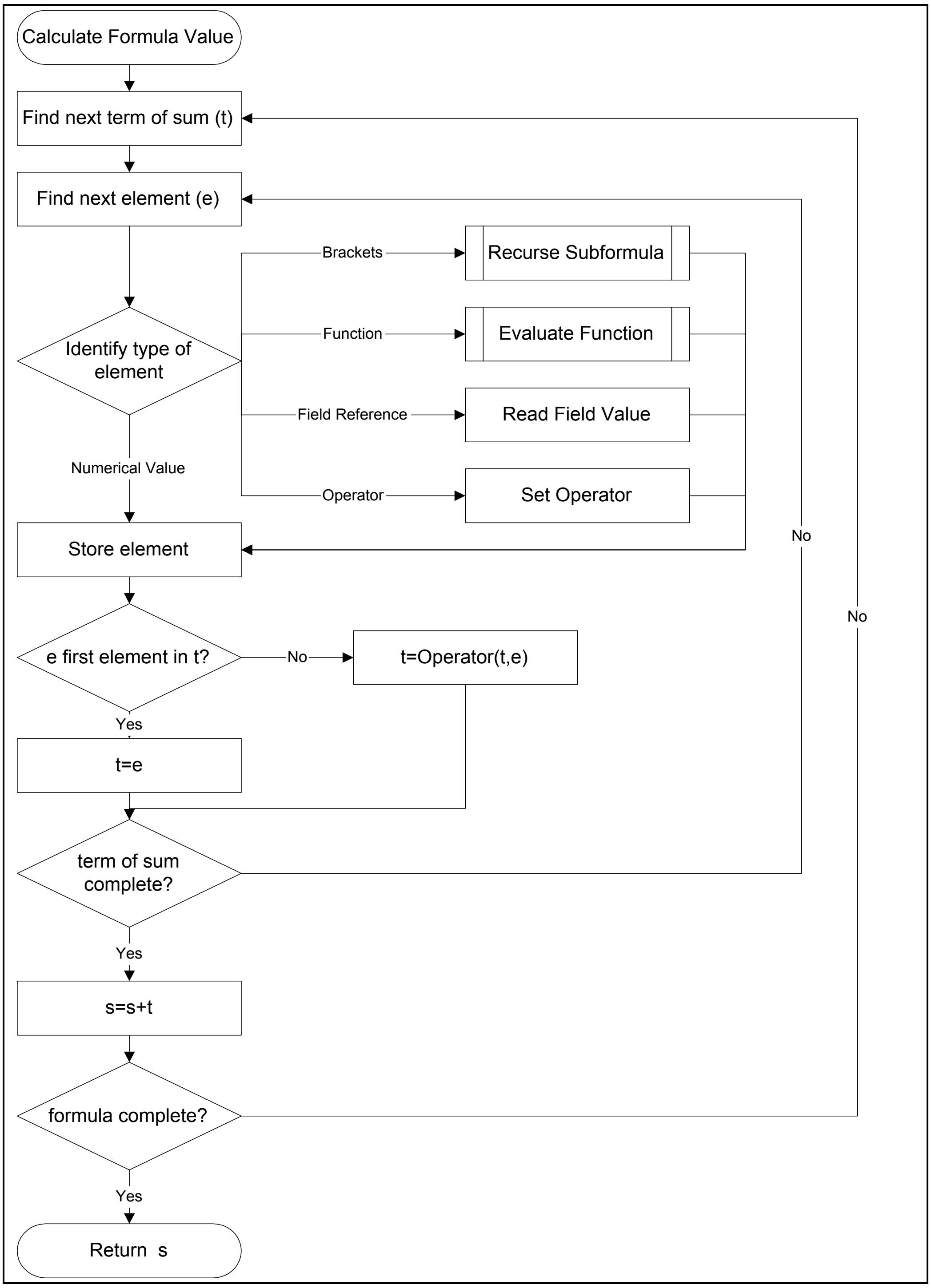
Figure 8-8 - House of Quality Calculation Example

For deploying the design characteristics in further matrices of the QFD process, the field [Rel Technical Weight] is often attached to the design items, because its values represents their relative importance to each other.

8.7.3 Formula Evaluation Algorythm

The implemented formula evaluation algorythm is performed recursive. A recursive algorythm is defined by being capable to recall itself.

The core of the algorythm is the *Calculate Formula Value* function. In this procedure the value of a particular field value is calculated by interpreting the formula defining it. A flowchart of the function is displayed in Figure 8-9.



Kommentar: formula3.vsd

Figure 8-9 - Flowchart of Evaluate Field Value

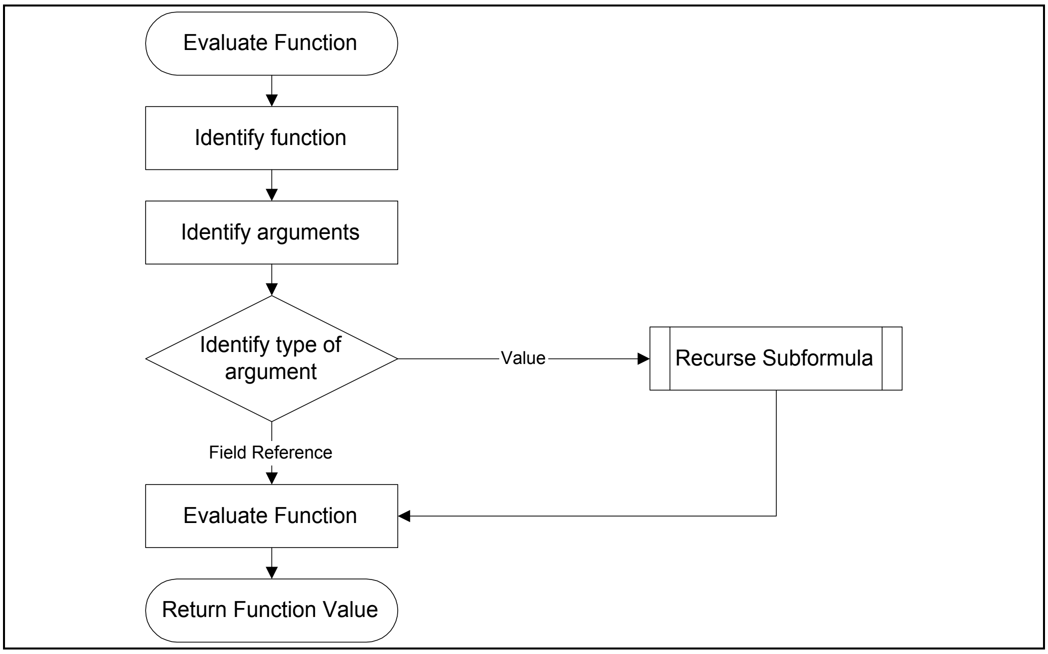
The procedure can be explained best by means of an example. A field may have to following formula definition:

= 2 * (abs ([field_a]) + 1)

The text of the formula definition is transfered to the *Calculate Formula Value* procedure. After removing unnecessary blanks, the first term of sum is seeked. In the example, there is only term of sum, which is the whole formula. After the term of sum is identified, the formula is splitted into elements that can be multiplied or divided by each other. The formula contains the elements "2" and "(abs([field_a]) + 1)" which are combined with the operator "*".

The first element "2" is evaluated and identified as a numeric value. Because it is the first value in the term of sum, it is stored directly into the temporary storage for the term of sum. The next element "(abs([field_a]) + 1)" is embraced in brackets. For that reason, it is recursed. Recursing means in this case, that the *Calculate Formula Value* procedure is recalled by itself for the evaluation of the formula "(abs([field_a]) + 1)" which is a subformula of the original formula. Each time a function is recursed, the level of recursion increases by 1, so we recurse to the second level of recursion now.

After removing the brackets, the recursion is committed. The subformula contains two terms of sum, "abs([field_a])" and "1". The evaluation of the first one leads to the identification as function. So the *Evaluate Function* procedure is called. A flowchart for this procedure is illustrated in Figure 8-10.



Kommentar: formula4.vsd

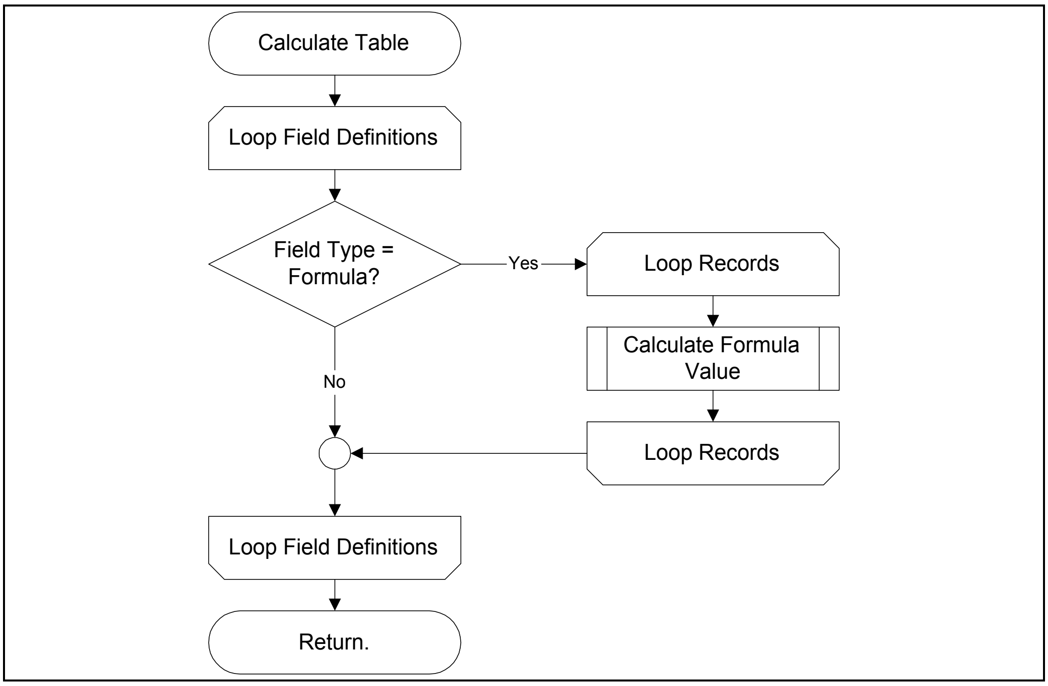
Figure 8-10 - Flowchart of Evaluate Function

The function in our example is identified as the *abs()* function, its argument is "[field_a]". The abs() function needs an argument of the type *value*. For that reason, we have to evaluate the content of the argument, which is again nothing else than a subformula that can be recursed in the *Calculate Formula Value* function.

On the third level of recursion the term "[field_a]” is identified as the only term of sum and the only element, containing a field reference. The respective value in *field_a* is read from the database, let’s say it was "-5”. This value is returned to the second level of recursion as argument for the abs() function.

The evaluation of the abs() cuts the minus of its argument, and returns "5” back to the *Calculate Formula Value* function on the second level. The term to be evaluated on the second level has now been reduced to "5+1”. After the sum is calculated to "6”, this value is returned to the first level, diminishing the original formula to a simple "2*6”. The second element ("6”) is multiplied with the first element ("2”), leading to the final field value "12”.

To calculate a whole database table, the procedure displayed in the flowchart of Figure 8-11 is used. For every field is tested if its type is *formula*. If so, the value of every record in the respective field is evaluated by recursing the *Calculate Formula Value* procedure.



Kommentar: formula2.vsd

Figure 8-11 - Flowchart of Table Calculation

In Figure 8-12 the flowchart of the overall matrix calculation is depicted. During the matrix calculation, one problem can occur: If a matrix() function is to be calculated in the horizontal table, the refered field in the vertical table that should contain the weighting values, is not evaluated yet. In that case, the vertical table has to be calculated *before* the horizontal part. If the vertical table contains a matrix() function as well, an error message is displayed. On the other hand, it is not necessary for methological reasons to calculate a matrix with matrix() functions on either side.

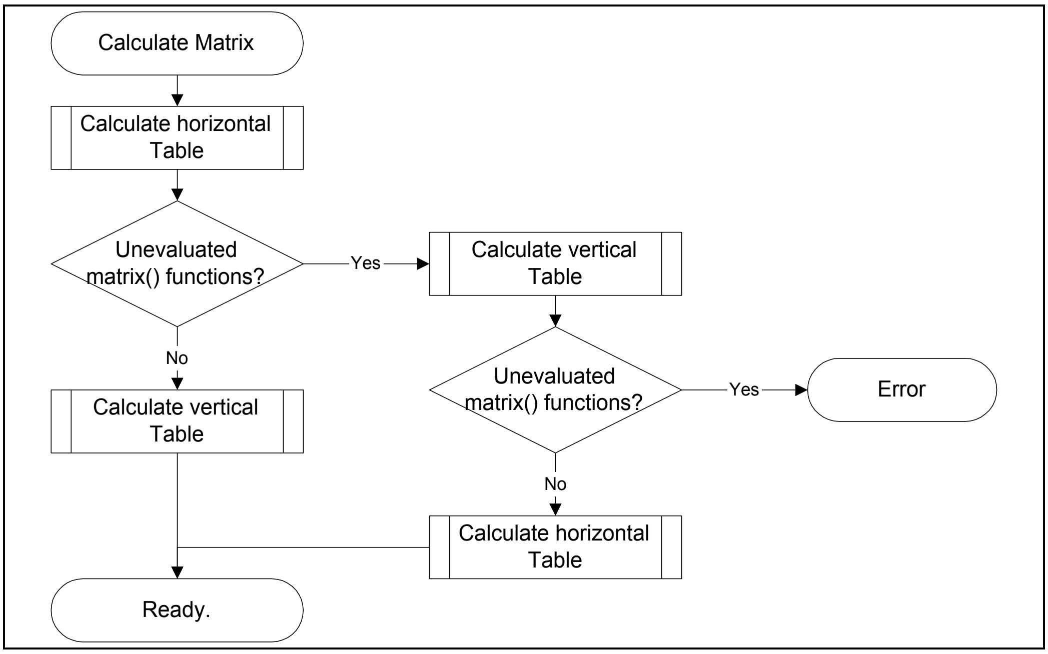


Figure 8-12 - Flowchart of Matrix Calculation

Kommentar: Formula1.vsd

The algorythm implemented in the QFD-Manager is far more complex than the described evaluation algorythm, but the basic procedure is the same. The reader is refered to QFD-Manager's Visual Basic source code.

8.8 Graphical Display and Matrix Print-Outs

8.8.1 The Need for Printing out Matrices

After the matrix data is gathered, entered, or calculated, all information is provided that forms a complete QFD matrix.

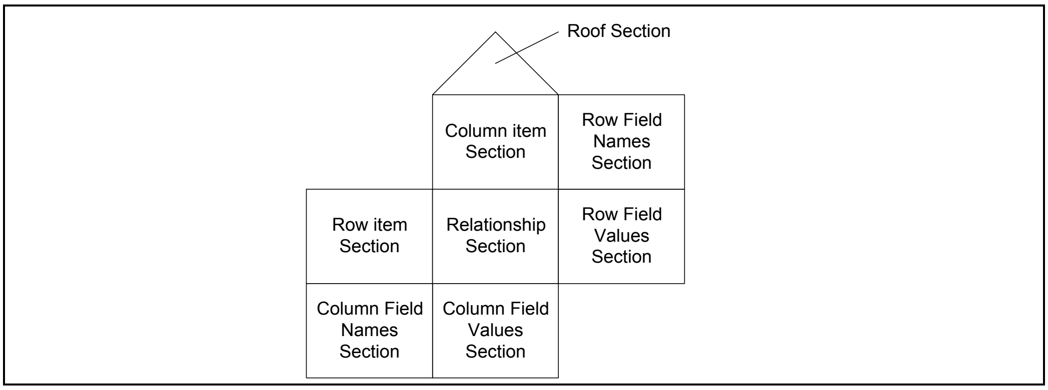
It is not sufficient to have the QFD data stored in the QFD database only. After a matrix is completed, it has to be distributed, discussed and verified by the team members. It is not only important to collect and provide the information given in the matrices, it has to be made usable for any member of the QFD team. For a QFD-System, it is therefore a crucial point to provide a function to print the matrix on a printer. If the design of a QFD-Support-System was pulled through a QFD process itself, the user want *"I want to print out my QFD data"* would be a *basic requirement*, which has to be necessarily provided for the success of the whole product. The user will also want to see what the final print out will look like before he starts printing. For that reason, a graphical screen output as a print preview is desirable. The user would also like to have a zoom function to take a more detailed look of certain parts of larger matrices.

The format of a QFD matrix makes it necessary to use a graphical interface for the display on the screen and on the printer. Although the text of the matrix items could still be printed out using a printer in text-only mode, major problems would arise while printing the horizontal and vertical matrix lines or the symbols used in QFD. The problem of printing out the matrix roof is not solveable with a text-only printer.

Visual Basic provides a graphical interface for windows on the screen and for all Windows compatible printers. The functionality is almost identical, for that reason some synergy effects can be used in the implementation of both screen and printer display.

8.8.2 Sections and Elements of QFD Matrices

A QFD matrix contains up to 6 sections. The matrix' two charts form a section for their items each (eg customer requirements or technical characteristics), a relationship matrix relating their items to each other, and room for additional data in their fields. The vertical chart can contain a co-relationship matrix, traditionally displayed as a roof over its items. For better distinction, the terms *item section*, *relationship section*, *field section*, and *roof section* will be used in the following (see Figure 8-13). The Field Section is further divided into the *Field Names* and the *Field Values* Section.



Kommentar: HOQ sections.vsd

Figure 8-13 - Sections of a QFD matrix

A matrix contains a set of elements. The different types of elements are *Text* (eg customer requirements, field names etc.), Numerical *Values* (eg field values like *"customer importance"*), *Symbols* (eg for relationships), and *Lines* to separate the other elements and to structure the matrix.

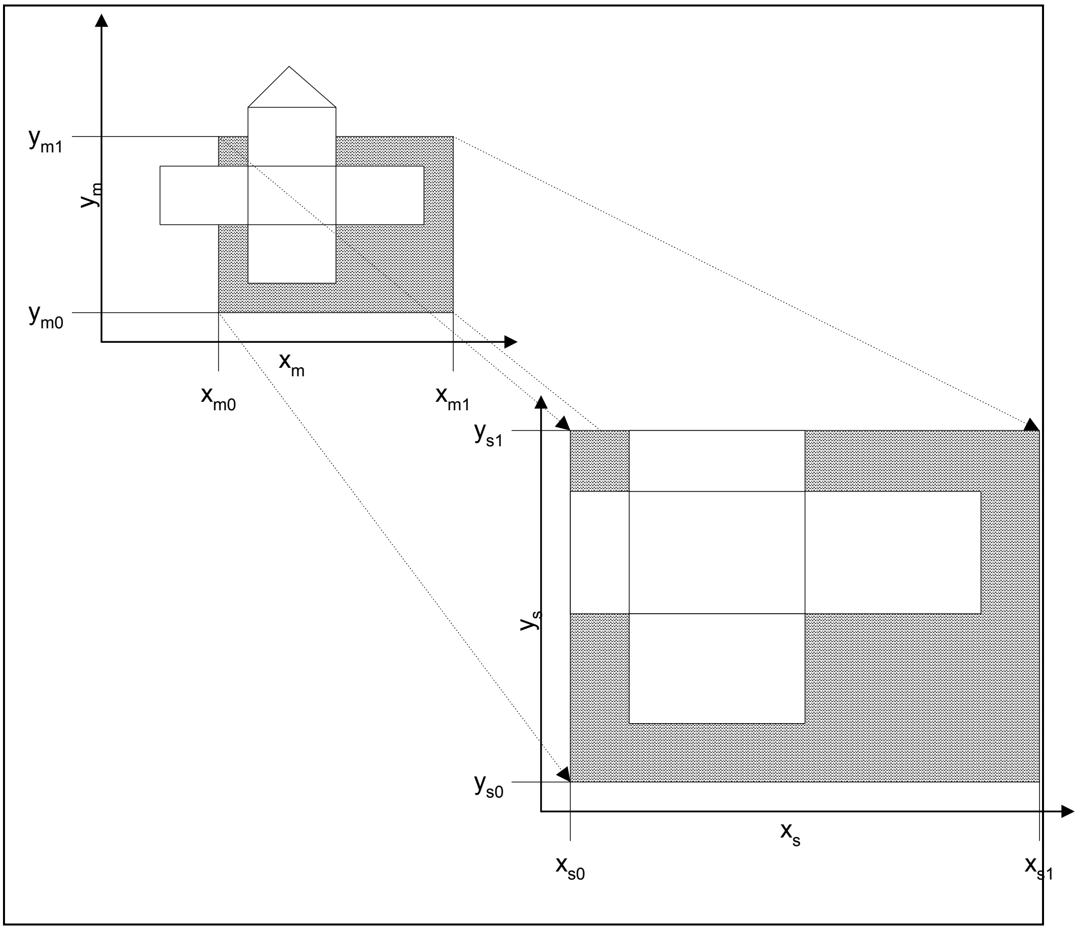
To display elements, the system needs information about the element and its position on the matrix. Beside the coordinates x_m and y_m of each element, values are needed to identify the maximum display space that is provided in the particular field of the matrix the element is printed in (see figure Figure 8-14).

text	text, x_m, y_m, $x_{m,size}$, $y_{m,size}$
value	value, x_m, y_m, $x_{m,size}$, $y_{m,size}$
symbol	identifier, x_m, y_m, $x_{m,size}$, $y_{m,size}$
line	line thickness, x_{m1}, y_{m1}, x_{m2}, y_{m2}

Figure 8-14 - Matrix Elements

8.8.3 Matrix, Screen and Printer Coordinate Systems

Printing the matrix on the screen or on the printer is basically a geometrical problem. We have to transform and project a certain coordinate system defining the matrix dimensions to the coordinate system of the screen or printer. The *screen coordinate system* and *printer coordinate system* are predefined by MS-Windows, the *matrix coordinate system* can be defined by us.



Kommentar: transform1.vsd

Figure 8-15 - Transformation from Matrix to Screen Coordinate System

The example in Figure 8-15 illustrates a coordinate system for a QFD matrix. The matrix is projected into another system displayed a part of the matrix. The example explains the linear transformation of a part of the matrix being displayed on the screen.

For the transformation, a ratio scale is needed. Presuming that the ratio of the x and y axis remains the same after transformation, one transformation ratio scale is sufficient. The ratio scale r_{ms} is defined as:

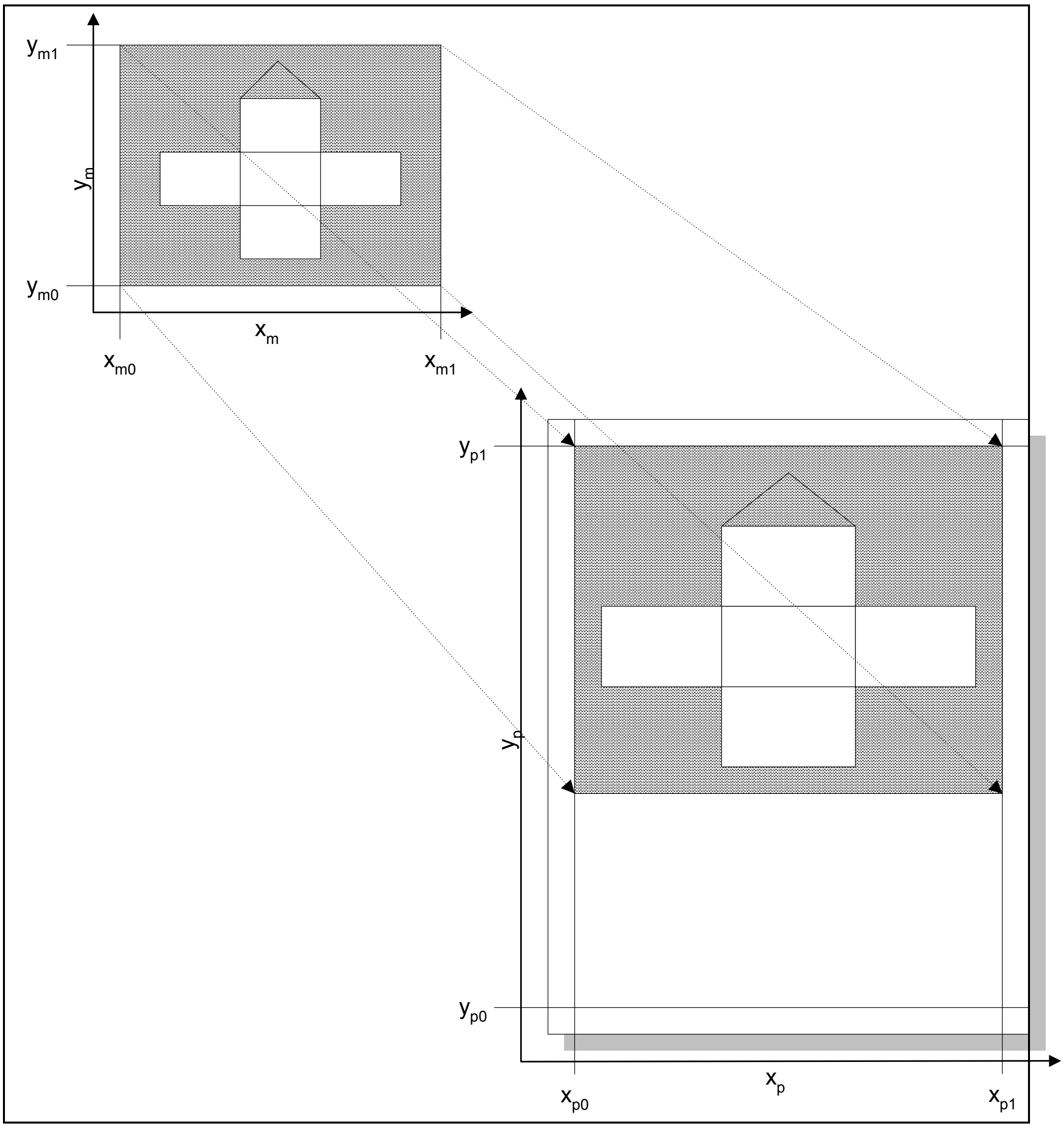
$$r_{ms} = (x_{s1} - x_{s0}) / (x_{m1} - x_{m0}) = (y_{s1} - y_{s0}) / (y_{m1} - y_{m0}) = dx_s / dx_m = dy_s / dy_m$$

To perform the transformation, the following formulas are used to project each point (x_m,y_m) from the matrix coordinate system to the refering point (x_s,y_s) of the screen system:

$x_s = x_{s0} + r_{ms} * (x_m - x_{m0})$

$y_s = y_{s0} + r_{ms} * (y_m - y_{m0})$

The same methodology applies for the transformation to the printer coordination system. In difference to the transformation to the screen system, we want to fit the matrix optimally on a printer page, that it can be printed on a single page and maximizes the available space of the sheet. For the printer transformation, the ratio scale r_{mp} between the matrix and printer coordinate system is defined as the minimum of the ratio $r_{mp,x}$ of the page width x_{p1}-x_{p0} to the matrix width x_{m1}-x_{m0} and the ratio of the page height y_{p1}-y_{p0} to the matrix height y_{m1}-y_{m0}. Choosing the minimum of both ratios ensures, that the matrix fits on a single page, even though some horizontal or vertical respectively space might be unused (see Figure 8-16).



Kommentar: transform2.vsd

Figure 8-16 - Transformation from Matrix to Printer Coordinate System

$r_{ps,x} = (x_{p1} - x_{p0}) / (x_{m1} - x_{m0})$

$r_{ps,y} = (y_{p1} - y_{p0}) / (y_{m1} - y_{m0})$

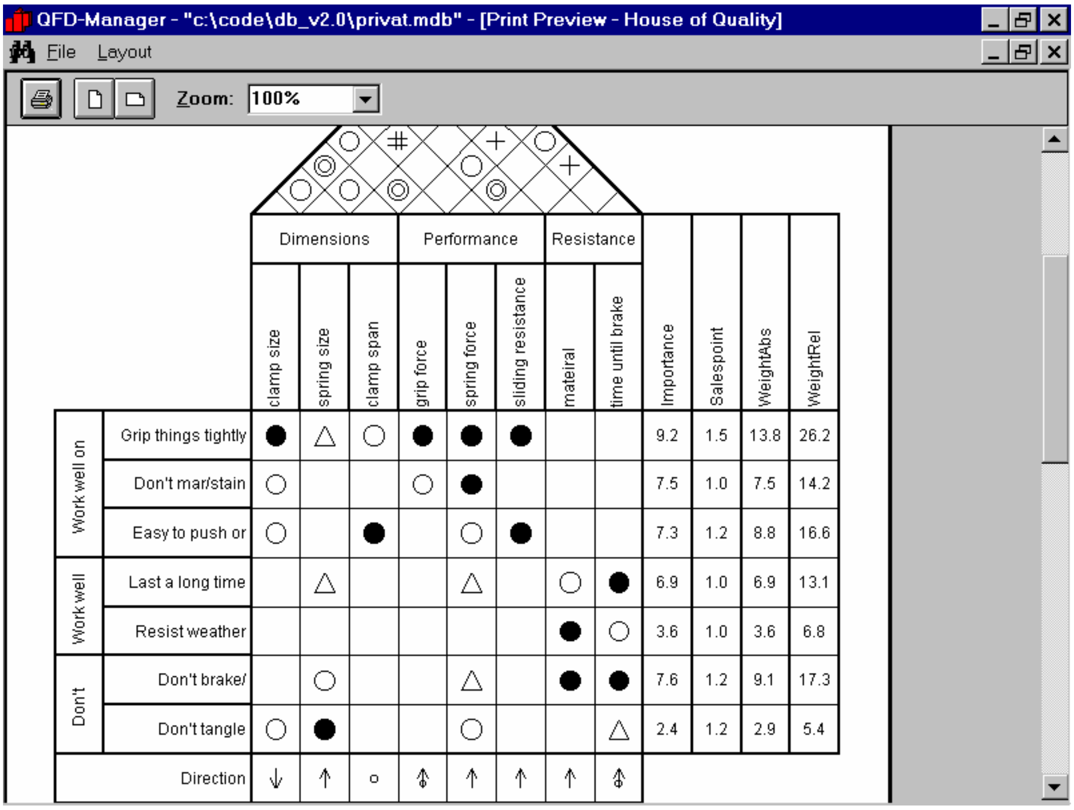
$r_{ps} = \min (r_{ps,x}, r_{ps,y})$

$x_p = x_{p0} + r_{mp} * (x_m - x_{m0})$

$y_p = y_{p0} + r_{mp} * (y_m - y_{m0})$

To display the matrix elements defined in Figure 8-14 on the screen and printer, their coordinates x_m and y_m are transformed to screen (x_s, y_s) or printer coordinates (x_p, y_p), which can be directly used to invoke system defined printing procedures.

A sample for a QFD-Manager print preview is illustrated in Figure 8-17.



Kommentar: Print Preview.bmp

Figure 8-17 - QFD-Manager Print Preview

8.8.4 Related Problems and Solutions

Providing the display functionality on the screen and on the printer leads to many problems during the implementation with Visual Basic. Although this is not the main topic of this thesis, a short overview over the encountered problems will be given.

- display the printer page on the screen

To gain a realistic print preview on the screen, the white printer paper has to be drawn on the screen. It was given a black "shadow" on the right and bottom to provide a three dimensional look.

- construct flexibility for different matrices

The logic of the internal data storage strucuture varies from the type of matrix the is to be displayed. Three different types of matrices have to be considered: horizontal charts (eg a pre-planning chart), vertical charts (eg a set of concepts) and *real* QFD matrices (eg House of

Quality). All different types have to be transformed in a different way into the screen or printer respectively coordinate system.

- Displaying the structure

To finally draw the matrix on the screen, a complex display procedure had to be applied. The procedure had to be flexible enough to cover all matrix variations possible in the QFD-Manager.

- Wordwrap, Textsize

A serious problem was the display of long text in the item and field text sections. When the text is longer as the horizontal (or vertical respectively) space in the matrix, the end of the text has to be wrapped into the next line. Due to the variable length of characters in the MS-Windows operating system, even the task of calculation the length of a string is not trivial. In case of wrapping, another problem occurs: The vertical (or horizontal respectively) alignment of the text changes and has to be adapted.

- Vertical Text

Visual Basic does not provide a function to print vertical text on the screen or on the printer. On the other hand, vertical text is vitally necessary to display QFD matrices due to the massive space comsumption of horizontal text eg in the vertical item section or the vertical field name section. To solve this problem, a user-defined font has to be created using some core functions of the operating system. To deal with this methodology, advanced knowledge about the operating system had to be acquired.

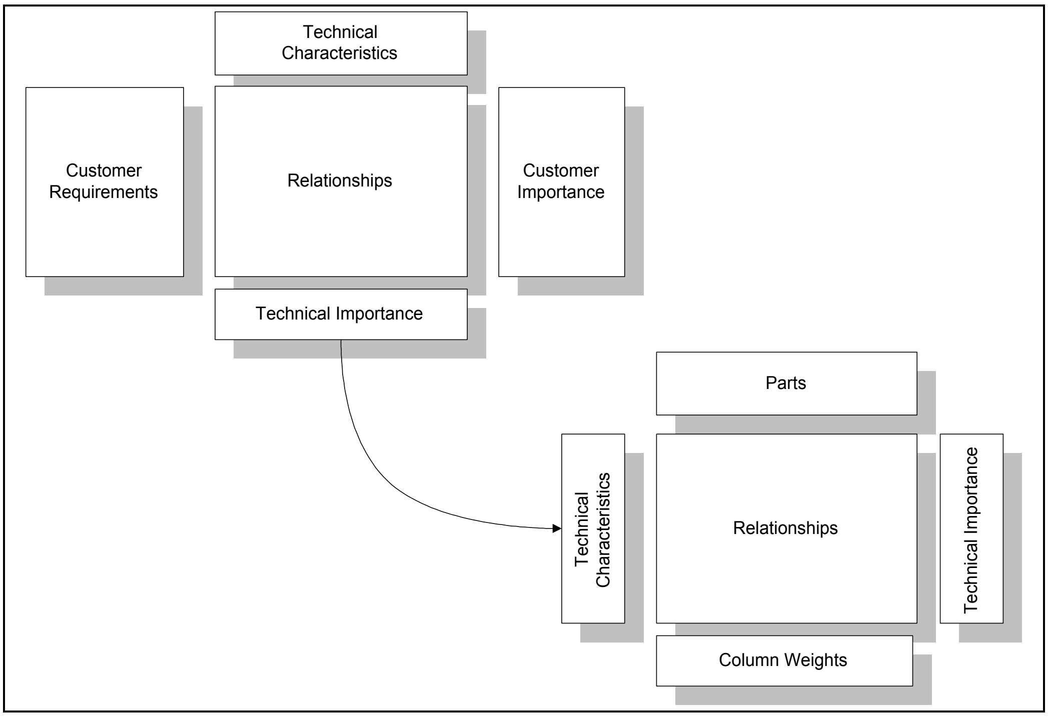
- Variable Cellsize and Context Menues

The size of section and sub-sections in the matrices have to be sizeable to provide a flexible matrix layout. To provide an advanced and user-friendly software, context menus were used to fulfill this task. The problem is to identify a underlying matrix cell by the coordinate of the mouse. Different Zoomfactors, roll bar positions, matrix dimensions and matrix cell sizes had to be considered.

Despite of the variety and seriousness of the encountered problems, a solution could be found for every item. A good source for obtaining help is the Usenet on the Internet with its *Visual Basic* news-groups.

8.9 Matrix Deployment

The third letter in QFD stands for "Deployment”. Datawise, deploying a QFD matrix means, to select some or all items of one chart (usually the vertical portion of the matrix) including some of their related data, and use this chart as input for another matrix (Figure 8-18). Repeating this method, QFD data can be deployed throughout the QFD process.



Kommentar: deployment.vsd

Figure 8-18 - Matrix Deployment

The initial input data of almost every QFD project are customer requirements, which form the horizontal part of the House of Quality. These customer requirements are deployed throughout the QFD process. For that reason the overall method is a *customer-driven* product development process.

The QFD-Manager provides a flexible and easy-to-use deployment function. The content of the respective matrix chart is deployed into a new chart. The fields to be deployed with the chart's items can be chosen in the Field-Manager by selecting the *Deploy* option. Beside the usual column deployment, rows can also be deployed. After the deployment, *formula* type fields are converted into *value* type fields, containing the formula results from the source chart.

8.10 Matrix Module Summary

The presented QFD-Manager Matrix Module provides a variety of powerful tools to handle QFD matrices and QFD charts.

The matrix data can be managed to branch to the Data-Manager. All mathematical matrix operations can be defined using a extremely flexible formula syntax. The matrix can be displayed on the screen using different zoom factors. A print-out can be produced in quality which is only limited by physical resolution of the printer. The QFD-Manager contains a deployment function, that deploys the data with related information into subsequent matrices. Both rows and columns can be deployed.

The modularity of QFD-Manager's Matrix Module yields a combined performance and flexibility which is unique compared to other QFD Support Systems currently available.

9 Concepts

A concept gives a brief description of the way a product is realised. The quality of a product's concept is crucial for the succes of the product.

In QFD the process of developing the best concept for a product is divided into two phases and begins with the *concept generation*. Different concept alternatives are collected using brainstorming techniques.

After that, the *concept selection* is performed. A concept evaluation is performed and the process is repeated until the team agrees about the best concept. The selected concept acts as the base for the subsequent development process.

9.1 Concept Generation

The generation of concepts is a creative process. There are many different methods to find new ideas that can lead to new concepts. Often used methods for concept generation are Trigger Word Technique, Checklist Technique, Morphological Chart Technique, Synectics Technique, or the wider known Brainstorming Technique. It is beyond the scope of this thesis to describe these methods in detail.

To foster creativity, the team can focus on different source of data. The sources can be divided into three classes[89]:

- Data from the QFD project
    - Project Mission Statement
    - Customer and Stakeholder Requirements
- Internal Sources
    - Experiences from previous products and development processes
    - Research and Development
    - Creativity of the QFD Team
    - Collected Ideas from other departments
- External Sources
    - Competitive Products on the Market
    - New Technologies and Knowledge

Pugh observed, that "engineers somtimes hold on tenaciously to their favourite engineering solutions".[90] Very often, team members seem to be constrained in their creativity because they think about existing solutions. To invent new concepts, these constrains have to be overcome. The feasibility of ideas should not be taken into account during the concept generation. Even the most questionnable concepts can integrate some solutions that could be used to develop other promising concepts. Smith proposes three basic rules for the success of idea generation[91]:

- Suspend judgement,
- Delay evaluation and give new ideas a change to grow,

[89] Westphal 1996, p.78

[90] Pugh 1981, p. 497

[91] Smith 1989, p.120, in: Westphal 1996, p. 79

- Idea screening and evaluation come later.

As stressed above, concept evaluation and filtering should generally be suspended during concept generation. But there are always some concepts that can be sorted out even during this phase. These are concepts that won't have the slightest chance to succeed in the concept selection, because they don't fulfill the *basic concept requirements*. These basic requirements include:

- Fulfill project mission,
- satisfy basic customer requirements,
- fulfill legal constraints,
- match industrial standards, and
- fulfill special company requirements.

Beside these crucial requirements, the evaluation of concepts is part of the next step, the concept selection.

In this chapter, Stuart Pugh's original example is used to give an idea about concept generation and selection. Figure 9-1 shows fourteen alternative designs for a car horn. The example shows the variety of concepts that can be invented just to make traffic more noisy.

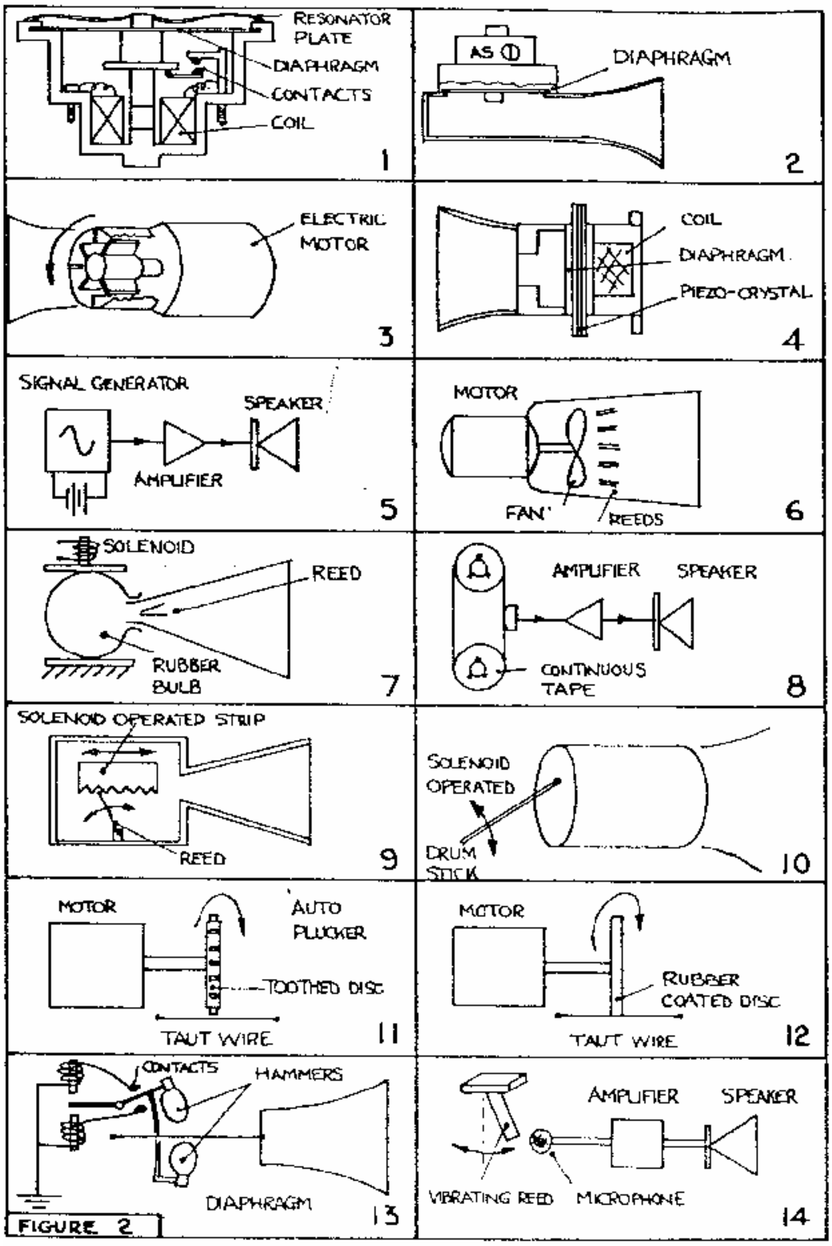


Figure 9-1 - Car Horn Concepts[92]

[92] Pugh 1981, p. 504

9.2 Computer Support for Concept Generation

The example illustrates, that concept generation is a creative process. The process follows no particular rules and is conducted each time in a different and unique way.

For that reason, computer support is at most useful for certain data management tasks. Conducting the concept selection process by using a computer will only lead to limitations in creativitiy and idea generation. Further, the software had to adapted to different circumstances each time.

Even if this problem could be solved, computer support would not provide any major benefits, because the overall amount of data is very small, compared to other tasks within the QFD process.

Because computer support for concept generation is neither recommendable nor feasable, special software tools for this task are not integrated in the QFD-Manager. Anyway, the list of concepts can be entered and modified into the project database by using the standard QFD-Manager database tools.

9.3 Concept Selection

Concept generation is a divergent process, which generates a couple of different design alternatives out of one design task. These concepts are evaluated in the following phase, the concept selection. The selection task is a convergent process, which evaluates the best design concept out of the variety proposed in the generation phase.

Selecting the right concept is crucial for the overall success of the new product. Pugh stresses, that especially when handling complex design problems, it is advanteous to deal with a couple of small decisions rather than with just a few major decisions. For that reason, he introduces a *concept comparison and evaluation matrix*, which is in current literature usually refered to as *Concept Selection Matrix*.

The Concept Selection Matrix is widely spread. It is used in a variety of QFD approaches, for instance in the King Modell, the CD-Map process, and in many others.

Concept	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Ease of achieving 105-125 DbA		S	-		+	-	+	+	-	-	-	-	S	+
Ease of achieving 2000-3000 Hz		S	S	*N*	+	S	S	+	S	-	-	-	S	+
Resistance to corrosion, erosion and water		-	-	*O*	S	-	-	S	-	+	-	-	-	S
Resistance to vibration, shock, acceleration	*R*	S	-	*T*	S	-	S	-	-	S	-	-	-	-
Resistance to temperature	*E*	S	-		S	-	-	-	S	S	-	-	S	S
Response time	*F*	S	-	*E*	+	-	-	-	-	S	-	-	-	-
Complexity: number of stages	*E*	-	+	*V*	S	+	+	-	-	-	+	+	-	-
Power consumption	*R*	-	-	*A*	+	-	-	+	-	-	-	-	S	+
Ease of maintenance	*E*	S	+	*L*	+	+	+	-	-	S	+	+	S	-
Weight	*N*	-	-	*U*	+	-	-	-	S	-	-	-	-	+
Size	*C*	-	-	*A*	S	-	-	-	-	-	-	-	-	-
Number of parts	*E*	S	S	*T*	+	S	S	-	-	+	-	-	S	-
Life in service		S	-	*E*	+	-	S	-	-	-	-	-	-	-
Manufacturing costs		-	S	*D*	-	+	+	-	-	S	-	-	-	-
Ease of installation		S	S		S	S	+	-	S	-	-	-	S	-
Shelf life		S	S		S	S	-	-	S	S	S	S	S	S
better		0	2		8	3	5	3	0	2	2	2	0	4
worse		6	9		1	9	7	12	11	8	13	13	8	9
total		-6	-7		7	-6	-2	-9	-11	-6	-11	-11	-8	-5

Figure 9-2 - Concept selection matrix[93]

Figure 9-2 shows the concept selection matrix for the car horn concepts. The concept selection criteria are listed in rows. The list of criteria can comprise many different aspects. In the example, technical characteristics and customer requirements from earlier stages of the QFD process are used, as well as design specifications and constrains, and safety and realibility issues.

The mix of criteria has a strong influence on the subsequent concept evaluation. Possible criteria include:

- Fulfillment of basic product functions,
- the degree of customer satisfaction it provides,
- the degree of achievemtn of target values set for technical characteristics in the technical table,
- reliability aspects, and
- safety issues.

All comparisons in the matrix are based on a reference product. The reference may be the best available product, or the company's current product.

---

[93] adapted from Pugh 1981, p. 505

Each concept is compared to this reference. For each criterion, the question is asked, "Is this concept better, worse, or equal to the reference?". In the matrix, a "+" represents better performance, "-" worse, and a "S" or a blank field means, that the performance is the *same* in both concepts. After all fields of the matrix are evaluated, the number of better, worse, and same performance is counted. The best concept is the one that shows the highest difference between the number of plus and minus values.

In the next step, the selected concept is used as reference, and the process is repeated until the team agrees about the final concept. During the repetitions, concepts may be varied, added, or removed, if its feasibility or performance proves not to be competitive at all. Concepts are refined, and additional data is collected during these subsequent phases. Pugh described, that it is not unusual that five or more evaluation phases are necessary to select the concept that is finally used for design, detailing, and manufacturing.

Pugh stresses two important points: The final selection must not be performed by only looking at the score on the concept selection matrix. The choice remains at the team's or management's responsibility. Many non-quantifiable aspects have to considered that can not be added to the concept selection matrix.

The second point is the non-numerical character of the matrix, which shall help to avoid false confidence in the calculated matrix score. The matrix is mainly a means to structurize the selection process and documents what exactly were the criteria for the final decision.

Although Pugh simple matrix is still the most widely spread approach to concept selection, the matrix has been refined since its introduction in 1981. Some approaches allow to enter numerical values into the matrix that differ from "1" and "-1" (plus and minus). Using these values, a more quantifiable comparison between the concepts can be achieved.

Other approaches introduce an importance column, that contain an importance value for each criterion. Using the values as multipliers for the plus/minus values in the matrix, a weighted score can be calculated. The calculation process is similar to the matrix calculation presented in figure Figure 8-1 of chapter 8.

A more sophisticated approach applies the Analytic Hierarchy Process (AHP, s. chapter 5.1.2.4) to evaluate the best concept[94]. Although the gained data has a higher quality compared to other approaches, the effort the perform AHP is very high, because for each criterion the concepts have to be compared with each other. Even Saaty, the inventor of the AHP, stresses, that the AHP is not feasible anymore when dealing with more than nine items to evaluate[95]. Considering that Pugh's original matrix even aimed to avoid a strong numerical character, the AHP approach seems to be rather scientific than practical.

9.4 Computer Support for Concepts Selection

As stated above, computer support for concept generation is not recommendable. On the other hand, concept selection is a more structured process containing a certain set of numerical or non-numerical data. Because it can be necessary to repeat the selection process a couple of times, computer support could be feasible and useful.

Aspects for possible computer support are:

- Handling of concept selection matrix
- matrix calculation and evaluation, and

---

[94] Kuppujaru, N. 1985

[95] Saaty 1996

- archiving concept data

A concept selection matrix is similar to ordinary QFD-matrices that were discussed in chapter 8. The two charts that necessary to define the matrix are the list of concept criteria (rows) and the list of concepts (columns). The relationship values between the criteria and the concepts are usually "1", "0", and "-1", represented by the symbols "+" and "-". The zero-value is represented by an "S" or a blank relationship field.

The QFD-Manager's Matrix-Module has been adapted to handle concept selection matrices. After defining the criteria, concepts and relationship tables, a *concept selection* option is chosen, which qualifies the Matrix as a concept selection matrix.

Subsequently, the Relationship-Manager is used to enter the concept values into the matrix. Instead of the traditional QFD matrix symbols, Plus/Minus symbols are displayed.

The calculation of the scores is different from ordinary QFD-matrices. For the evaluation, additional functions are required that count the number of plus or minus values in the columns of the matrix. To maintain the high flexibility of the Matrix Module, these functions were designed to be compatible with the formula definition and calculation process that was described in chapter 8.7. The following functions were added to the set of matrix calculation functions:

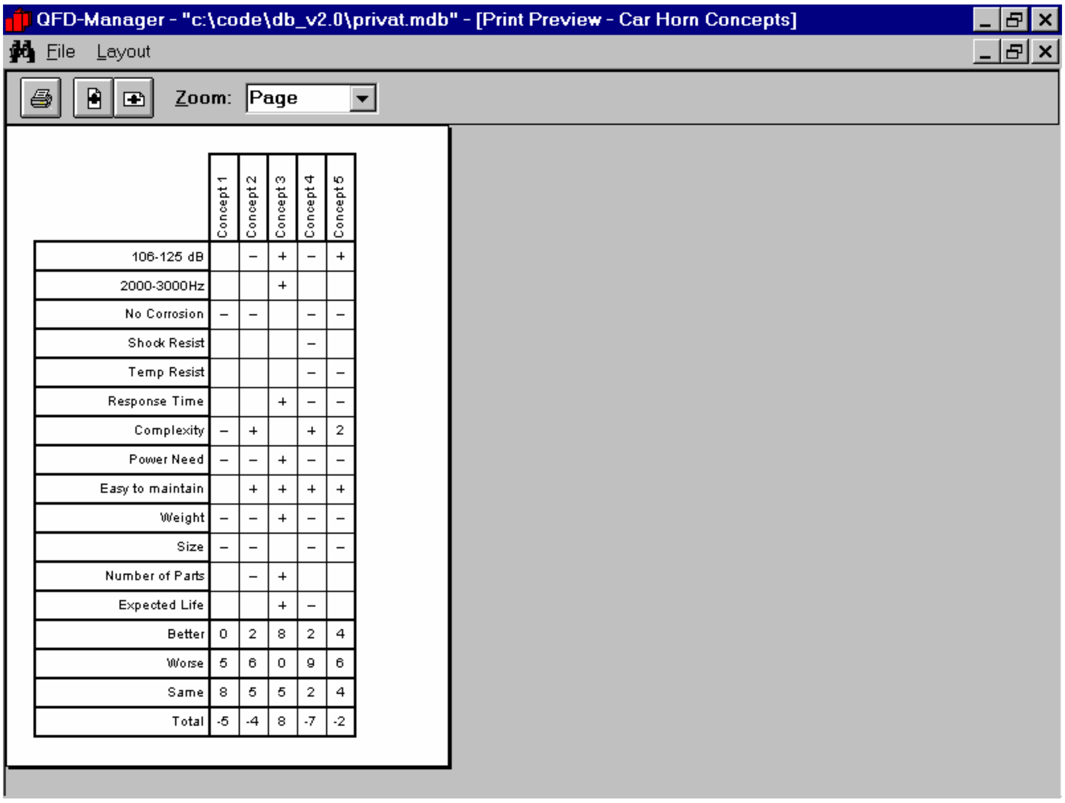
pugh()	Returns the column sum
pughpos()	Returns the sum of the column's positive values
pughneg()	returns the sum of the column's negative values
pughsame()	returns the number of the column's zero values

Figure 9-3 - Concept Functions

The pugh() functions provide a flexible and powerful way to evaluate concept selection matrices. The functions need no argument. Figure 9-4 shows an example for the application of the functions. The last four rows of the matrix are calculated using the pughpos(), pughneg(), pughsame(), and pugh() function.

To give an example for values that differ from the symbol values 1 and -1, the last concept is evaluated regarding the criterion "Complexity" by a value of "2". This value influences the pughpos() function, which summarized the positive column values to "4". The pughpos() and pughneg() function don't only return the *number* of positive values, but the *sum* of these values. This methodology provides the highest degree of flexibility for the concept evaluation process, because not only pure pugh-matrices can be evaluated, but also matrices that use numerical values instead of plus/minus symbols. As shown in the example, even *mixed* concept selection matrices are possible with the QFD-Manager.

Although the value of the plus/minus symbols (1 and -1) will not have to be changed very often, this feature is also provided in the QFD-Manager.



	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5
106-125 dB		–	+	–	+
2000-3000Hz			+		
No Corrosion	–	–		–	–
Shock Resist				–	
Temp Resist				–	–
Response Time			+	–	–
Complexity	–	+		+	2
Power Need	–	–	+	–	–
Easy to maintain		+	+	+	+
Weight	–	–	+	–	–
Size	–	–		–	–
Number of Parts		–	+		
Expected Life			+	–	
Better	0	2	8	2	4
Worse	5	6	0	9	6
Same	8	5	5	2	4
Total	-5	-4	8	-7	-2

Figure 9-4 - Concept Selection Matrices in the QFD-Manager

But the QFD-Manager provides even more flexibility. By applying the Field-Manager on the Criteria Table, an importance field can be added. This field can be filled with importance values for each criterion. The relationships values, even when they are different from 1, 0, or -1, can now be *weighted* before they are summed up to the total concept score. In this case, the calculation would be performed with the matrix() function, which is used to calculate a *weighted column sum* (see chapter 8.7).

After the calculation, the matrix is printed out for all stakeholders within or outside the QFD team. The paper serves as base for the following discussion. If the selection process has to be repeated, all subsequent data handling can be done easily within the QFD-Manager using its deployment functionality (see chapter 8.9).

Figure 9-5 illustrates a possible concept deployment process. The concept table (Concepts 1) is deployed from the concept selection matrix into a new concept table (Concepts 2). This table forms the base for the second concept selection phase. Before the evaluation is conducted, the list of concepts can be extended, diminished, or modified. The criteria from the first stage can be applied in the subsequent phases by defining them as part of the subsequent concept selection matrices. The process is continued until the team agrees about the final concept.

Especially in the concept deployment process, the QFD-Manager provides a degree of flexibility that is unique compared to other QFD-Systems. Beside its flexibility, it provides a comprehensive solution for handling QFD concept data.

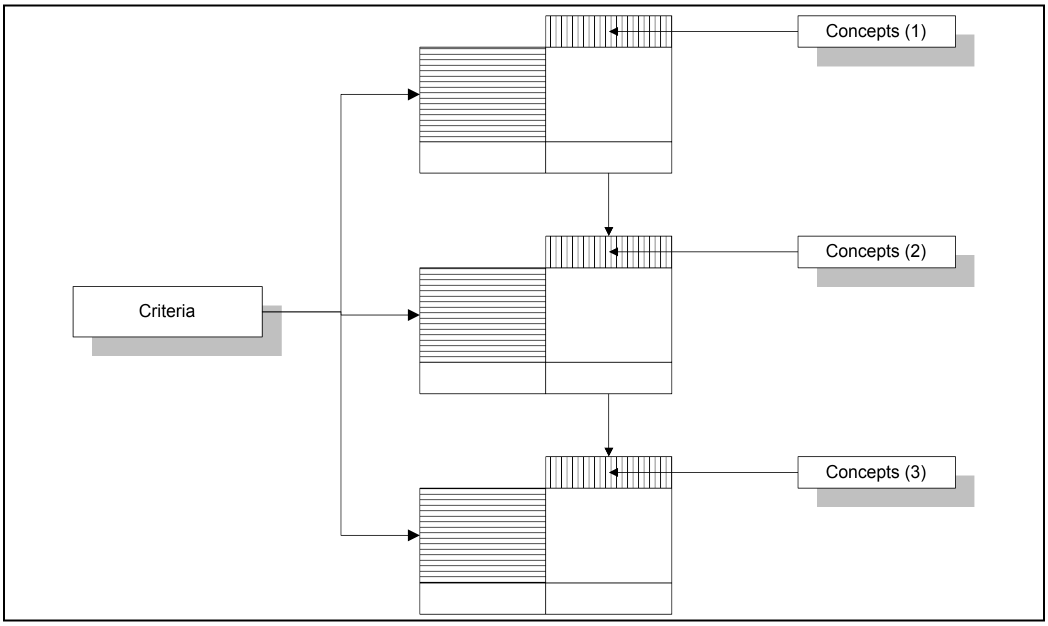


Figure 9-5- Concept Selection Process using QFD-Manager's Deployment Functionality

10 System Overview

In the preceding chapters, the different modules of the QFD-Manager and their data structures were discussed. This chapter provides an overview of the entire data structure of the system. Before the structure of the database tables on the logical level is discussed, the conceptual database will be described using an Entity-Relationship diagram.

10.1 Conceptual Database Design

The conceptual database is an abstraction of the *real world* that focuses on elements of information that are relevant to the user of the database. To display conceptual databases, Entity-Relationship (ER) diagrams are used (see chapter 2.5). Using ER diagrams, complex structures can be visualized in a simple and standardized way. They don't represent the underlying table structure of the database, which is represented by the logical database design (see chapter 10.2).

In Figure 10-1, the entire ER diagram of the QFD-Manager is displayed. It represents the conceptual database design of the QFD-Manager. The diagram forms the base for the underlying data structures and their relations that were discussed in the preceding chapters.

The core of QFD-Manager's Entity Relationship diagram is the *Container* which is displayed as the large box with the rounded corners on the right. A database table which is classified as Container comprises *Records*, which are displayed left of the Container. Records have at least the attributes *ID*, *Text*, *Comment*, *Level*, *Parent*, and *Position*, which form the basic structure of a Record.

A Record can comprehend more attributes than there are in the basic structure. These additional attributes can be defined in QFD-Manager's Field-Manager. Certain predefined constellations of additional attributes represent certain sub-entities of Records. The three types of sub-entitites are *Requirement*, *Characteristic*, and *Concept*. These sub-types form the data structures that are required for special modules in the QFD-Manager, as for instance the *VOC Merging Module* and *VOC Data Tools* which require the *Requirement* structure, and the Characteristics-Module, which are based on *Requirement* and *Characteristic*. The three types of sub-entities can for a *Requirement Set,* a *Characteristics Set*, or a *Concepts Set*, which are sub-entitites of the entity *Container*. A *set* comprehends a list of *records*.

The principle of a constant basic structure and flexible additional attributes yields strong advantages. On the one hand, data compatability with all general modules of the QFD-Manager is achieved. Each entity that is based on the container structure can be used in the Affinity-Module, Survey-Module, Matrix-Module, Data-Manager etc.. On the other hand, an entity based on the container structure is highly flexible and can carry any QFD related information. Not only the explicitly mentioned sub-types *Requirement*, *Characteristic* and *Concept* are possible, any other data type can be defined, eg. Functions, Cost Information, or sets of criteria for special evaluations.

The entities *Raw Statement* and *Customer Voice* are used for the Requirements Generation process (see chapter 3.7). They contain the extracted Raw Data from the Extraction Module, which is merged into the masterfile. The relation between *Raw Statement* and *Requirement* shows, that a requirement can be derived from none to four customer statements (raw statements), and a raw statement can lead to none or more requirements. This illustrates the requirements conversion process of the VOC Data tools.

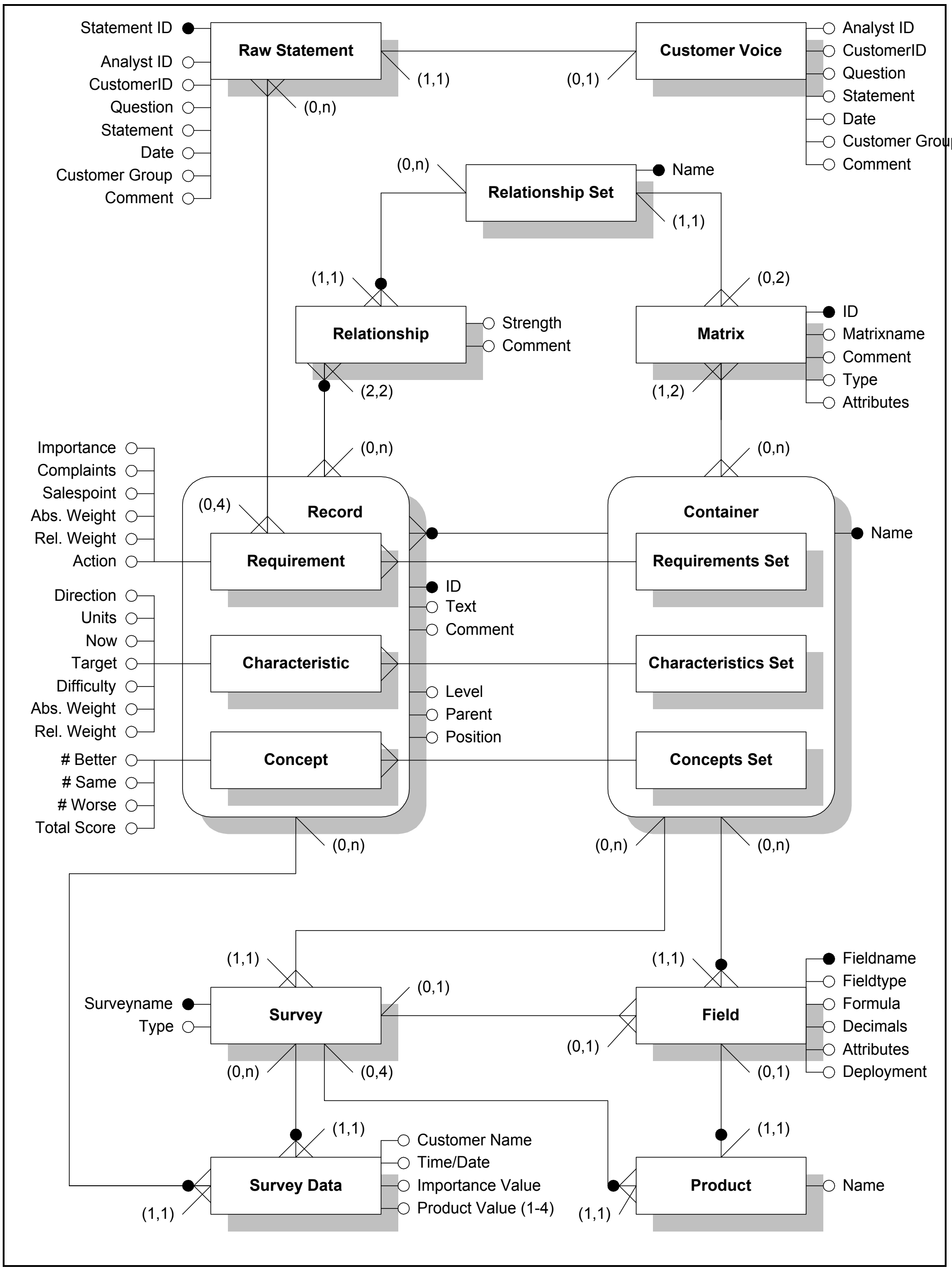


Figure 10-1 - QFD-Manager Entity-Relationship Diagram

Kommentar: ER_QFD-Manager.vsd

A QFD matrix definition is represented by the entity *Matrix*. As discussed in chapter 8.3, a matrix is defined as a combination of one or two containers, up to two sets of relationships, and some additional information. In the entity relationship diagram, *Matrix* relates to the entity *Container* with the cardinality (1,2). The minimun cardinality of one is necessary, because a matrix in the QFD-Manager terminology can comprise only one container, which acts eg as a pre-planning chart. Beside this special case, two containers are usually assigned to a matrix.

A matrix can refer to *Relationship sets*. Relationship Sets comprehend a certain set of *Relationships* which relate records with their numerical attribute *Strength*. Relationships and Relationship Sets are used to represent the content of relationship matrices, co-relationship matrices, the values of concept selection matrices, and any other matrices defined in the Matrix Module. A matrix can relate to up to two relationship sets, which represent each the used relationship and the co-relationships set of the matrix. Although a container can be used in more than one matrix, a relationship set can only be part of one matrix.

The lower part of the Entity-Relationship diagram represents the data structures of the Survey Module and their relations to the main structure of the QFD-Manager. A *Survey* is based on exactly one *Container*. Usually the Container will have to structure of a *Requirements Set*, but other types are suitable for conducting a survey as well. *Surveys* can relate to a *Field* that carries the importance values, and up to four *Products* for the competitive survey. The ranking value of *Products* is stored in *Fields* as well. A *Field* is always assigned to exactly one *Container*.

Below *Survey*, the entity *Survey Data* is displayed. *Survey Data* comprises all information retrieved from the customer. It relates to a survey and to a particular *Record*, usually a *Requirement*.

10.2 Logical Database Design

The translation of the conceptual database yields the logical database. The logical database is a set of tables which comprise fields and records. The tables can be connected logically by means of relations. The logical database is the actual description of the database structure that acts as base for the QFD-Manager.

The tables of the QFD-Manager's database have been presented in the preceding chapters, where the individual modules of the system were discussed. In this chapter, an overview of the different database tables is given which relates to the translation from the ER diagram.

The basic table containing QFD information is the *Container Table*. The Container Table can be used by all general modules of the QFD-Manager, such as the Affinity-Module, the Survey-Module, and the Matrix-Module.

ID	Text	Comment	Level	Parent	List Pos.
long int.	text (100)	text (100)	integer	long int.	integer

Figure 10-2 - Container Table

Based on the Container Table, the container sub-types *Requirements Table*, and *Characteristics Table* are constructed. The following figures illustrate the compulsory fields to identify the particular sub-type, and some additional fields that are added automatically by the QFD-Manager (*italic font*). The additional fields can be edited or removed using the Field-Manager. For display reasons, the basic structure of the Container table is displayed as field type *Basic*.

Container Base	Sources	Source1	Source2	Source3	Source4
basic	long int.	long int.	long int.	long int.	long int.

Importance	*Complaints*	*Salespoint*	*WeightAbs*	*WeightRel*	*Action*
single	single	single	single	single	text

Figure 10-3 - Requirements Table

Container Base	Direction	Measurement	*Company Now*	*Target*	*Difficulty*	*WeightAbs*	*WeightRel*
basic	long int.	long int.	single	single	single	single	single

Figure 10-4 - Characteristics Table

The fields Source1 to Source4 in the Requirements Table points to records in the *Masterfile* which contains the customer statements from the extraction process. The Masterfile has to following structure:

StatementID	AnalystID	CustomerID	Question	Statement	Date	Customer Group	Comment
long int.	long int.	long int.	text(15)	text(100)	date	integer	text (100)

Figure 10-5 - Masterfile Table

The *Matrix Definition Table* contains the attributes of a matrix and pointers to the underlying container and relationship tables (last four fields).

ID	Matrix name	Comment	Type	Attribute s	Row table	Column table	Relations. table	Co-rel. table
long int.	text (64)	text (255)	long int.	long int.	text (64)	text (64)	text (64)	text (64)

Figure 10-6 - Matrix Definition Table

The *Relationship Table* connects to records of a container table with a certain relationship value (strength).

ID1	ID2	Strength	Comment
long int.	long int.	single	text

Figure 10-7 - Relationship Table

The following figure shows the structure of *Survey Definition Table*. Next to the survey name, a reference to the container table is included which forms the base for the input and output data of the survey process. *Importance Field* contains the field of the container table which will carry the surveyed importance value of each container item. To simplify the table structure for the surveys, the entity *Product* is integrated into the Survey Definition Table. Each product has a name and a reference field. Up to four products can be included into the competitive survey.

ID	Survey Name	Tablename	Survey Type	Importance Field
long integer	text	text	integer	text(64)

Product Field 1	Product Name 1	Product Field 2	Product Name 2	Product Field 3	Product Name 3	Product Field 4	Product Name 4
text(64)	text	text(64)	text	text(64)	text	text(64)	text

Figure 10-8 - Survey Definition Table

The retrieved survey data is contained in the Survey Data Table. Each record contains information about the importance values, the competitive ranking, the survey it refers to (*Survey ID*), and the container item that is ranked. The table is the base for the survey evaluation function of the QFD-Manager.

ID	SurveyID	Customer Name	Date/ Time	Record ID	Importance	Product 1	Product x...
long integer	long integer	text	text	long integer	numeric	numeric	numeric

Figure 10-9 - Survey Data Table

A closer description of the database tables is given in the respective chapters of this paper.

Kommentar: xyz: Capabilities and Bounds of the System - Function Analysis of the System

10.3 Future Developments

The presented system provides a great variety of helpful support functions for QFD projects. More important, the system provides a base for further enhancements because of its generic and flexible data structures. Other functions can be included into the system without great efforts of changing the current data structures and related functions.

New aspects which could be introduced into a enhanced system comprise:

- Function Analysis.

A Function Analysis is an important part of many QFD projects. Functions Analyses can be conducted using the FAST process.[96] The FAST process can be divided into two tasks: The first task helps to create functions, in the second task the functions are structured into a hierarchical structure.

[96] Fowler 1990, pp. 77ff

Westphal suggests a "question and answer" module which could help to conduct the functions-generation task.[97] Although this approach is not followed in this thesis, it could be implemented into a enhanced version of the system.

- Voice of the Customer Table

No attempt is made in this paper to implement support for the *Voice of the Customer Table* (VOCT). This field is still to be researched in further theses.

- Failure Analysis

Westphal also suggest to implement the failure analysis system into the system which was proposed by Satoshi Nakui[98]. Conducting this analysis, obvious mistakes in QFD matrices can be spotted automatically. This function can be implemented easily into the QFD-Manager - for example as a function which can be called from a menu of the Matrix-Module (eg. *Matrix Failure Analysis*). The spotted errors could be displayed on the screen, printed out or stored into a file. Changes in the matrix could be suggested automatically.

- Sensitivity Analysis

Some suggestions for a sensitivity analysis in QFD can be found in current literature. To implement this approach into a computer system, more sophisticated methods and data structures have to be applied. The implementation of a sensitivity analysis is clearly beyond the scope of this paper.

- Analytic Hierarchy Process

The Analytic Hierarchy Process (AHP) has not been implemented into the Quantitative Survey Module due to its massive time consumption for the respondent. Anyway, it is possible to implement the AHP into the system. The necessary changes comprise the media creation procedures and media retrieval functions.

- Enhancement of Graphical Display

Due to development time and ressource restrictions, the QFD-Manager suffers from a less powerful and flexible matrix display on the screen and as print-out. Further enhancement are possible.

For example, the matrix fields of the competitive analysis is often displayed in a graphical way in current literature[99]. This could also be performed of the Matrix-Module of the QFD-Manager without greater changes in the respective modules. Further, the user friendlyness of the related functions could be improved.

- QFD/DES interface

The QFD/DES interface suggested by the International Techne Group International (ITI) and used in the software QFD/Capture could be implemented into the matrix module. This task is quite easy to realise as the interface structure is defined entirely in the QFD/Capture manual. Both import and export interface are easy to design by setting up a conversion procedures that connects to the QFD/DES file and the QFD-Manager project database. The interface could also be extracted from the QFD-Manager as a stand-alone program.

---

[97] Westphal 1996, p. 129ff

[98] Nakui 1992

[99] see eg. Day 1993, p. 53

11 Summary

In this paper a computer software for supporting QFD projects is developed. The new system is called *QFD-Manager* and is based on the QFD-Support-System of Ingo Westphal.[100]

The presented system is not just an enhancement of the old system, it is rather based on a completely new system and data structure. It provides both flexibility and performance to support QFD tasks. The system can be run as a stand-alone system or over a network by connecting to a centralised database.

The main drawback of commercial QFD software currently available on the market is its limitation to simply "draw matrices". The QFD-Manager provides not only means of entering data into matrices and print them out, but also helps to generate the data. The key customer phrases from the qualitative survey can be extracted and stored into a database. The phrases can then be converted into customer requirements. The list of requirements can be structured using the Affinity Process. The requirements can be translated into design characteristics which form the vertical portion of the House of Quality. The design characteristics can be structured in a similar way to the requirements as well.

Based on the final list of customer requirements the quantitative survey can be performed. Three different types of survey media are offered including paper questionnaires, diskettes and the Internet. Using the latter two, the survey can be conducted with the highest degree of automation offered by computer technology currently available.

As in commercial QFD software packages, the QFD data can be arranged into matrices. The flexibility offered by the QFD-Manager is very high: All widely spread types of QFD matrices can be used including standard QFD matrices, concept selection matrices and pre-planning charts. Data in the matrix can be calculated using a flexible and easy-to-use formula language. The matrices can be printed out on every printer supported by the MS-Windows operating system. Also implemented is a deployment function which deploys matrix data into new matrices. Both columnwise and rowwise deployment is possible. Using this flexible approach the project is not constrained to the Four-Phases model (Macabe model) like most of the commercial QFD software, but can even used for complex QFD deployments as the Akao model, all variations of the King model and Enhanced QFD. The QFD-Manager is a generic tool for covering a vast majority of QFD projects.

The open structure of the system allows to implements enhancements where necessary. This is achieved by the definition of a highly flexible data structure that carries structured lists of QFD data as customer requirements, design characteristics, functions and concepts. This data can be used in all modules of the system.

The presented system is capable of showing methods to support QFD projects efficiently. It handles QFD data and supports the time-consuming matrix calculations, the matrix display and print-outs. Further, it improves communication among the QFD team members by providing always the latest data on the system.

The system could be used as a template for a professional version which could be introduced on the QFD software market.

---

[100] Westphal 1996

12 References

Adiano, C. (1994), Dyanmic QFD - Beyond the House of Quality, in: The 6th Symposium on QFD, Novi, Michigan, USA, 1994

Akao, Yoji (1990), QFD: Integrating Customers Requirements into Product Design, Cambridge, Massachussetts, USA, 1990

Batini, C. (1992), Conceptual Database Design, Redwood City, California, USA, 1992

Blumstein, Gershon (1996), Why QFD fails and what to do about it, in: The 8th Symposium on QFD, Novi, Michigan, USA, 1996

Brassard, Michael (1989), Memory Jogger II - Users Manual, Methuan, MA, USA, 1989

Codd, E.F. (1970), A Relational Model of Data for Large Shared Data Banks, in: CACM, V13, N6, June 1970, p. 377ff, , 1970

Dahlheimer, Per (1995), The Customer Input Process, Sydney, 1995

Day, Ronald G. (1993), Quality Function Deployment - Linking a Company with its Customers, Milwaukee, Wisconsin, USA, 1993

Fowler, Theodore C. (1990), Value Analysis in Design, New York, 1990

Gardarin, G. et al. (1989), Relational Databases and Knowledge Bases, Reading, Massachusetts, USA, 1989

Goodman, John (1989), The Nature of Customer Satisfaction, in: Quality Progress, February 1989, pp. 37-40, , 1989

Griffin, Abbie (1992a), Evaluating QFD's Use in US Firms as a Process for Developing Products, in: Journal of Product Innovation Management 9/1992, pp. 171-187, , 1992a

Griffin, Abbie; Hauser, John R. (1992b), The Voice of the Customer, in: Massachusetts Institute of Technology, Report No. 92-106, Cambridge, Massachusetts, USA, 1992b

Guinta, Lawrence R. (1993), The QFD Book, New York, 1993

Gustafsson, A.; Gustafsson, N. (1994), Exceeding Customer Expectations, in: The 6th Symposium on QFD, Novi, Michigan, USA, 1994

Hales, R. et al. (1991), Electronic Exchange of QFD Data, in: The 3rd Symposium on QFD, Novi, Michigan, USA, 1991

International TechneGroup Inc. (1991), QFD/Capture Users Manual, Version 2.2, Milford, Ohio, USA, 1991

Kano, N. et al. (1984), Attractive Quality and Must-Be Quality, in: Hinshitsu 14, No.2, Japanese Society for Quality Control, Tokyo, 1984

King, Bob (1989), Better Designs in Half the Time: Implementing QFD in America, Methuen, Massachusetts, USA, 1989

Klein, Robert L. (1990), New Techniques for Listening to the Voice of the Customer, in: The 2nd Symposium on QFD, Novi, Michigan, USA, 1990

Kuppujaru, Nagesh et al. (1985), Design through Selection: A Method that Works, in: Design Studies, Vol. 6, No. 2/1985, , 1985

Lyman, D. (1990), Deployment Normalization, in: The 2nd Symposium on QFD, Novi, Michigan, USA, 1990

Nakui, Satoshi (1992), Gaining a Strategic Advantage: Implementing Proactive Quality Function Deployment, in: The 4th Symposium on QFD, Novi, Michigan, USA, 1992

Parsaye, Kamran et al. (1989), Intelligent Databases, New York, 1989

Pugh, Stuart (1981), Concept Selection - A Method that Works, in: International Conference on Engineering Design, 9-13 March 1981, pp. 497-506, Rome, Italy, 1981

Putzbach, I. (1995), Recent Trends and Developments in QFD for Product Design and Development, Sydney, 1995

Saaty, Thomas L. (1977), A scaling Method for Priorities in Hierarchical Structures, in: Journal of Mathematical Psychology, Vol. 15, No. 3, June 1997, pp. 234-281, , 1977

Saaty, Thomas L. (1980), The Analytic Hierarchy Process, New York, 1980

Saaty, Thomas L. (1996), Decision Making for Leaders, Pittsburgh, Pennsylvania, USA, 1996

Smith, N. et al. (1989), Managing for Innovation, London, 1989

Stauffer, Larry A. et al. (1996), A Computerized Database to Assist QFD, in: The 8th Symposium on QFD, Novi, Michigan, USA, 1996

Sullivan, L.P. (1986), QFD - A system to Assure that Customer Needs Drive the Product Design and Production Process, in: Quality Progress, June 1986, , 1986

Terninko, John (1995), Step by Step: Customer Driven Product Design, Nottingham, New Hampshire, USA, 1995

Urban, G.L. et al. (1993), Design and Marketing of New Products, 2nd edition, New Jersey, USA, 1993

Westphal, Ingo (1996), An Investigation into a Computer-Based Support System for Customer-Driven Product Development with QFD, Sydney, 1996

Yoder, Brian; Sosenko, Jessica (1994), Prioritisation of Customer Wants through the Use of a Pre-planning Matrix, in: The 6th Symposium on QFD, Novi, Michigan, USA, 1994

Zultner, Richard E. (1990), Software QFD, in: The 2nd Symposium on QFD, Novi, Michigan, USA, 1990

Zultner, Richard E. (1991), Before the House - The Voice of the Customer in QFD, in: The 3rd Symposium on QFD, Novi, Michigan, USA, 1991

Zultner, Richard E. (1993), Priorities: The Analytic Hierarchy Process in QFD, in: The 5th Symposium on QFD, Novi, Michigan, USA, 1993