Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Computers & Industrial Engineering 48 (2005) 427–441
computers & industrial engineering
www.elsevier.com/locate/dsw

Group decision making to better respond customer needs in software development

Gülçin Büyüközkan*, Orhan Feyzioğlu

Department of Industrial Engineering, Galatasaray University, Çırağan Cad. No: 36 Ortaköy, İstanbul 34357, Turkey

Abstract

Quality Function Deployment (QFD) is a well-known planning methodology for translating customer needs into relevant design and production requirements. The intent of applying QFD is to incorporate the voice of the customer into the various phases of the product development cycle for a new product, or a new version of an existing product. The traditional QFD structure requires individuals to express their preferences in a restricted scale without exceptions. In practice, people contributing to the process tend generally to give information about their personal preferences in many different ways, numerically or linguistically, depending on their background. Moreover, collaborative decision-making is not an emphasized issue in QFD even though it requires several people's involvement. In this study, we extend the QFD methodology by introducing a new group decision making approach that takes into account multiple preference formats and fusing different expressions into one uniform group decision by means of fuzzy set theory. An application on software development is supplied to illustrate the approach.
© 2005 Elsevier Ltd. All rights reserved.

Keywords: Quality function deployment; Group decision making; Fuzzy sets; Software development

1. Introduction

In a highly competitive environment, it is essential to be continuously ready and fully functional in order to cope with the market change and innovation, and to satisfy the customer needs. In this setting, product development (either physical or service) is an extremely important and strategic tool for the company. It encompasses a large variety of activities ranging from idea generation to final product delivery and requires not only an enhanced technological knowledge but also an effective quality management. The key success factor in quality is the rigorous customer orientation. It involves a deep

* Corresponding author. Tel.: +90 212 227 4480; fax: +90 212 259 5557.
E-mail addresses: gbuyukozkan@gsu.edu.tr (G. Büyüközkan), ofeyzioglu@gsu.edu.tr (O. Feyzioğlu).

0360-8352/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2005.01.007

understanding about who are the customers and what are their intentions, what are the key technologies that can satisfy these needs, and also requires a careful analysis about who are the competitors and what they perform in the same context. The objective must be to go beyond the customers' expectations by means of innovation and creativity (Wallace, 1992).

Effective customer requirement management is also crucial to the development of large-scale software systems. Unfortunately, it also a least understood issue (Gong, Yen, & Chou, 1998; Issac, Rajendran, & Anantharaman, 2004). Problems in the requirements are typically not recognized until late in the development process, where negative impacts are substantial and cost for correction has grown large (Davis, 1993; Dromey, 2003). Even worse, problems in the requirements may go undetected through the development process, resulting in software systems not meeting users' expectations. Therefore, methods and frameworks helping software developers to better manage software requirements are of great interest (Karlsson, 1997; Kudikyala & Vaughn, 2005).

Quality Function Deployment (QFD) is a comprehensive quality system aimed at satisfying the customer. It is a proven tool that is applied in many industries including software development since many years (Chan & Wu, 2002; Herzwurm & Schockert, 2003). The intent of applying QFD are to incorporate the voice of the customer into the various phases of the product development cycle for a new product, or a new version of an existing product. In this study, we suggest a new group decision making approach with multiple expression formats for QFD with the hope to better capture and analyze the demand of customers. We also show how these different expressions can be aggregated into one collaborative decision by using fuzzy set theory. Although applied in software development process, the presented approach is flexible enough to be used within the development of other products as well.

The rest of the paper is organized as follows. Section 2 describes briefly the QFD process and its use in software development. In Section 3, we present in details a new fuzzy logic based group decision making method for QFD. An illustrative example on word processing software development is then given in Section 4. Finally, Section 5 contains some concluding remarks and perspectives.

2. Quality function deployment and software development

QFD is a comprehensive quality tool specifically aimed at satisfying customers' requirements. It was originated in the late 1960s to early 1970s, in Japan, by Prof. Yoji Akao (1972). QFD is defined as a method and technique used for developing a design quality aimed at satisfying the consumer and then translating the consumer's requirements (CRs) into design requirements (DRs) and major quality assurance points to be used throughout the production stage (Akao, 1990).

The QFD process involves four phases:

(1) product planning: house of quality (HOQ),
(2) product design: parts deployment,
(3) process planning, and
(4) process control (quality control charts).

A chart (matrix) represents each phase of the QFD process. The complete QFD process requires at least four charts, called houses, to be built that extend throughout the entire system's development life cycle and to clearly establish relationships between company functions and customer satisfaction.

These matrices explicitly relate the data produced in one stage of the process to the decisions that must be made at the next process stage. In the product planning matrix, CRs in customers' own words (WHATs), are determined and translated into DRs (HOWs) or proposed performance characteristics of the product. The second QFD matrix relates potential product features to the delivery of performance characteristics. Process characteristics and production requirements are related to engineering and marketing characteristics with the third and fourth matrices. The more emphasized activity in QFD is the accurate prioritization of requirements. This is due to the fact that the requirements are of different importance for a given stakeholder group (users, developers, customers, environmental regulations, company policies, standards, etc.) and as expressed earlier, the focus is on the voice of the customer. The majority of QFD applications stop with the completion of the first matrix (or HOQ). Many companies, such as Volvo, have found that a great deal of benefit can be achieved from just completing the first matrix (Han, Chen, Ebrahimpour, & Sodhi, 2001). Similarly, Cox (1992) indicates that no more than 5% of companies go beyond the first matrix. For this reason, we only focalize in this study to the first matrix of the QFD structure. However, the proposed framework may be easily generalized for the remaining QFD stages.

The term software includes many systems, such as basic software operating systems, large-scale on-line systems for the different industries, application systems like computer-aided design, and others for many complex functions. Software quality has traditionally been defined in terms of fitness for use (Dunn, 1988). A software product is deemed to be fit for use if it performs to some level of user satisfaction, in terms of functionality and continuous operation (O'Brien, 1991). In order to achieve high customer satisfaction and high feasibility, the customers' requirements should be carefully specified and analyzed. Requirements analysis of a software system is often considered as one of the most crucial steps in the software development process (Liu, 1998), in which statements describing the functions and characteristics of the forthcoming software system should be developed and agreed upon. Software requirements are divided into two separate categories (Karlsson, 1997):

(a) The functional requirements are the core of the statement, describing the services that are expected by the stakeholders. Functional requirements typically describe the relationships between all valid (and invalid) inputs to the software system and the corresponding outputs from the software system.
(b) The non-functional requirements are often used to determine all characteristics a software system must possess that cannot be described as functional requirements. For example, usability and reliability describe characteristics pertinent to the actual usage, whereas requirements like schedule and cost estimates describe characteristics pertinent to the actual development of the software system.

In the requirement analysis process, all stakeholders develop a common understanding of the requirements describing the software system to be developed. The process can consist of the following stages: requirements discovery, documentation, validation, implementation selection and deployment.

QFD is one of the techniques that can overcome problems occurred in this process by especially focusing on the voice of the customer. It has proved its usefulness in software development (Basili & Musa, 1991; Elboushi & Sherif, 1997; Haag, Raja, & Schkade, 1996; Herzwurm & Schockert, 2003; Herzwurm, Schockert, & Mellis, 2000; Liu, 2001; Richardson, 2001; Zhou, 1998; Zultner, 1992). Haag (1992) describes how a number of major corporations have adapted QFD to document user specifications. Large development organizations such as Hewlett Packard, IBM, and DEC have used this tool to take advantage in improving the quality of applications delivered to customers (Haag, 1993). An overview of the state-of-the-art of QFD in software development is also given recently in Herzwurm and Schockert (2003). Experiences show that QFD is particularly beneficial in: (1) clear understanding of user requirements, (2) interdisciplinary communication, (3) development of consensus about the solutions found, (4) reducing the number of post-delivery changes, (5) documenting all steps taken, (6) creating a profitable product and (7) satisfying customers (Herzwurm & Schockert, 2003).

Companies attempting to implement QFD have reported a variety of benefits and also problems with the method (Bouchereau & Rowlands, 2000; Govers, 2001). Several attempts have been made to overcome the difficulties in carrying out the QFD process. Two of these trends are considered in this study: application of the fuzzy set theory and group decision making techniques to determine priorities of CRs and DRs in QFD.

QFD process may involve various inputs in the form of linguistic data, which are inherently vague, and it is a fact that human perception, judgment and evaluation on importance of CRs are usually subjective and uncertain. This case can be treated to approximate exactness with the help of fuzzy set theory (Zadeh, 1965). As a result, it has been widely used in QFD (Bouchereau & Rowlands, 2000; Erol & Ferrell, 2003; Harding, Popplewell, Fung, & Omar, 2001; Liu, Jia, & Viswanathan, 1998; Masud & Dean, 1993; Verma, Smith, & Fabrycky, 1999). An overview of the particular fuzzy logic applications in QFD is recently given in Chan and Wu (2002).

Group decision-making is another important concern in QFD. Multiple decision makers are often preferred rather than a single decision maker (Chiclana, Herrera, & Herrera-Viedma, 1998; Herrera, Herrera-Viedma, & Chiclana, 2001) to avoid the bias and minimize the partiality in the decision process (Lee & Kim, 2000). This was already suggested in Hales (1995), Ho, Lai, and Chang (1999) and Lai, Ho, and Chang (1998) where authors reveal the need for effective group decision-making techniques for dealing with QFD's team-oriented characteristics. Moreover, people tend to give information about their personal preferences in many different ways, numerically or linguistically, depending on their background. The expression of the preference information in different formats has also attracted many researchers (Herrera et al., 2001; Xu, 2004; Zhang, Chena, & Chong, 2004). Preference information in QFD consists of subjective descriptions of an individual or a group of people's desires about how they would like the product to be (Dubois & Prade, 2004). In this study, we propose a group decision-making approach in which the prioritization of customer requirements in QFD is achieved by fusing multiple expression formats in one collaborative group decision with fuzzy logic. The Section 3 is devoted to the details of this approach.

3. Fuzzy group decision making approach for QFD

In order to facilitate the description of the proposed approach, the following assumptions and notation are used:

- CRs are determined in hierarchical levels: we denote the first level requirements as $\{r^m: m=1,\ldots,M\}$, the second level requirements as $\{r^{mn}: m=1,\ldots,M;\ n=1,\ldots,N_m\}$, etc.
- The customers are categorized into K focus groups and each group member is denoted as $\{e^{kl}: k=1,\ldots,K;\ l=1,\ldots,L_k\}$ where L_k is the size of the group k. The importance weight (according to the target market) of each focus group is determined whereas members within a group are treated equally.

- The relationship strengths between CRs and DRs are expressed in the usual scale: 9, strong; 3, moderate; 1, weak. Occasionally, other scale values are used. The literature, however, does not document any basis for choosing the value set (Sivaloganathan & Evbuomwan, 1997).

At first stage of the approach, our aim is *to make the information obtained from group members uniform*. Let us assume that an individual e^{kl} gives his/her importance value on N ($R=\{1,\ldots,N\}$) CRs according the following formats:

1. e^{kl} can give an ordered vector (o(1),…,o(N)) where o(i) is the importance ranking (the more important is 1 whereas the least important is N) of CR i. This importance ordering can be transformed into relative importance relation as follows:

$$p_{ij} = 9^{u_i - u_j}, \quad 1 \leq i \neq j \leq N. \tag{1}$$

where $u_i=(N-\mathrm{o}(i))/(N-1)$.

2. e^{kl} can give an importance degree vector $(u_1,\ldots,u_N)$ where $u_i \in [0,1]$ $i=1,\ldots,N$. The importance degree u_i will more significant if it is close to 1. This vector can be transformed into relative importance relation such that:

$$p_{ij} = \frac{u_i}{u_j}, \quad 1 \leq i \neq j \leq N. \tag{2}$$

3. e^{kl} can give a linguistic importance vector $(s_1,\ldots,s_N)$ where s_i $i=1,\ldots,N$ can be one of 'Not Important (NI)', 'Some Important (SI)', 'Moderately Important (MI)', 'Important (I)' and 'Very Important (VI)'. One possible set of membership functions for these linguistic terms can be found in Fig. 1. Among all the different types of fuzzy numbers, the choice of triangular ones is made for the sake of simplicity, since assuming more complicated shapes may increase the computational complexity without substantially affecting the significance of the results. As for the modeling issue, it has to be recalled that Ma, Kandel, and Friedman (2000) procedure can be used to extract, from any type of fuzzy number, the nearest symmetrical triangular number which reflects most of the information

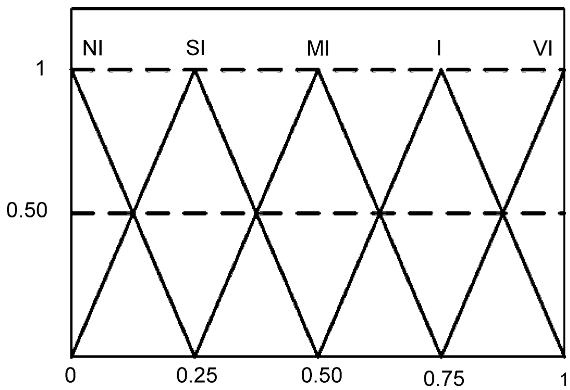

Fig. 1. Fuzzy numbers for 'importance to customers'.

contained in the original number. Therefore, the main implications of the present model would not be substantially affected by the choice of a different type of number but from a computational viewpoint.

Given that a fuzzy triangular number s_i can be noted as (a_i,b_i,c_i) where b_i is the most encountered value, the linguistic term vector can be transformed into relative importance relation as follows:

$$p_{ij} = \frac{b_i}{b_j}, \quad 1 \le i \ne j \le N. \tag{3}$$

4. e^{kl} can prefer to select only a subset of CRs ($\bar{R}$) and argue that they are important for him/her. For this case, the CRs in set $\bar{R}$ are equivalent to each other and dominate those in $R/\bar{R}$. The CRs in $R/\bar{R}$ are also equivalent to each other. The preference relation can be defined as:

$$p_{ij} = \begin{cases} 9, & \text{if } i \in \bar{R}, j \in R/\bar{R} \\ 1/9, & \text{if } i \in R/\bar{R}, j \in \bar{R} \quad 1 \le i \ne j \le N \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

5. e^{kl} can prefer to select only a subset of CRs ($\bar{R}$) and give the importance of those requirements linguistically. Using the same notation given in 3, the relative importance relation can be defined as:

$$p_{ij} = \begin{cases} 9^{b_i - b_j}, & \text{if } i,j \in \bar{R} \\ 9^{b_i - 0.5}, & \text{if } i \in \bar{R}, j \in R/\bar{R} \quad 1 \le i \ne j \le N \\ 1, & \text{if } i,j \in R/\bar{R} \end{cases} \tag{5}$$

6. e^{kl} can express that some CRs are more important than some others without explicitly identifying the degree. In this case:

$$\begin{aligned} &p_{ij} = 9 \text{ and } p_{\text{ji}} = 1/9 \text{ if } i \text{ is more important than } j \\ &p_{ij} = 1 \text{ if nothing mentioned} \end{aligned} \tag{6}$$

7. e^{kl} can give a pair wise comparison matrix where each element represents the relative importance of one CR compared to another. This can be achieved by using the ratio scale originally proposed by Saaty (1980): $p_{ij}=1$ if CR i and j are equally important, while $p_{ij}=9$ CR i is absolutely much more important than j. Intermediate importance values range from 2 to 8. The matrix is multiplicatively reciprocal, in other words:

$$p_{ij} = a \quad \text{and } p_{ji} = 1/a \text{ for all } a \in \{1, \dots, 9\} \tag{7}$$

It can be observed that Eqs. (1)–(7) enable us to make the information obtained from the customers uniform. In a second phase, *individual assessments are gathered to detect group opinion*. This process will reflect the opinions of the majority of customers. On this line, let $\{p_{ij}^{k1}, p_{ij}^{k2}, \ldots, p_{ij}^{kL_k}\}$ be the set of values to be aggregated for any $i,j \in R$ and group k customers. In our case, the ordered weighted geometric (OWG) operator of dimension L_k is a function $\phi^{\mathrm{G}} : \Re^{L_k} \to \Re$ which is associated a set of weights W and is defined as

$$\phi^{\mathrm{G}}\left(p_{ij}^{k1}, p_{ij}^{k2}, \ldots, p_{ij}^{kL_k}\right) = \prod_{l=1}^{L_k} (\bar{p}_{ij}^{kl})^{w_l} \tag{8}$$

where $W = (w_1, \ldots, w_{L_k})$ is an exponential weighting vector, such that $w_l \in [0,1]$ and $\sum_l w_l = 1$, and each $\bar{p}_{\mathrm{ij}}^{kl}$ is the l largest valued element in the set $\{p_{ij}^{k1}, p_{ij}^{k2}, \ldots, p_{ij}^{kL_k}\}$ (Herrera et al., 2001; Zhang et al., 2004).

The OWG operator reflects the fuzzy majority calculating its weighting vector by means of a fuzzy linguistic quantifier according to Yager (1988) ideas. Traditionally, the majority is defined as a threshold number of individuals. In this study, we make use of the fuzzy majority which is a soft majority concept expressed by a fuzzy linguistic quantifier. Two types of linguistic quantifiers can be distinguished: *absolute* and *proportional* (Zadeh, 1983). Absolute quantifiers are used to represent amounts that are absolute in nature such as *about* 2 or *more than* 5. These absolute linguistic quantifiers are closely related to the concept of the count or number of elements and for any given $r \in \Re_+$, the membership degree $Q(r)$ of r indicates the degree to which the amount r is compatible with the quantifier represented by Q which is a fuzzy subset. Proportional quantifiers, such as *most*, *at least half*, may be represented by fuzzy subsets of the unit interval, [0,1]. Then for any $r \in [0,1]$, $Q(r)$ indicates the degree to which the proportion r is compatible with the meaning of the quantifier it represents. For a non-decreasing relative quantifier, Q, the weights are obtained as

$$w_l = Q(i/m) - Q((i-1)/m), \quad l = 1, \ldots, L_k \tag{9}$$

where Q is defined as

$$Q(y) = \begin{cases} 0, & \text{if } y < a \\ (y-a)/(b-a), & \text{if } a \le y < b \\ 1, & \text{if } y \ge b \end{cases} \tag{10}$$

with a, b, $y \in [0,1]$ and $Q(y)$ indicating the degree to which the proportion y is compatible with the meaning of the quantifier it represents. Some examples of the relative quantifiers (see Fig. 2) are 'most' (0.3, 0.8), 'at least half' (0, 0.5) and 'as many as possible' (0.5, 1).

When the fuzzy quantifier Q is used to calculate the weights of the OWG operator ϕ^{G}, it is represented by ϕ_Q^{G}. Therefore, the collective multiplicative relative importance relation is obtained as:

$$p_{ij}^{k} = \phi_Q^{\mathrm{G}}\left(p_{ij}^{k1}, p_{ij}^{k2}, \ldots, p_{ij}^{kL_k}\right) \quad 1 \le i \ne j \le N \tag{11}$$

In a third phase, *the group opinion* collected in matrix P^k obtained through Eq. (11) *must be exploited to determine the importance weights* of the CRs. Note that in matrix P^k, the ij element reflects the relative importance of CR i compared to the requirement j. We will now calculate the quantifier guided importance degree of each CR, in other words, we will try to quantify the importance of one CR

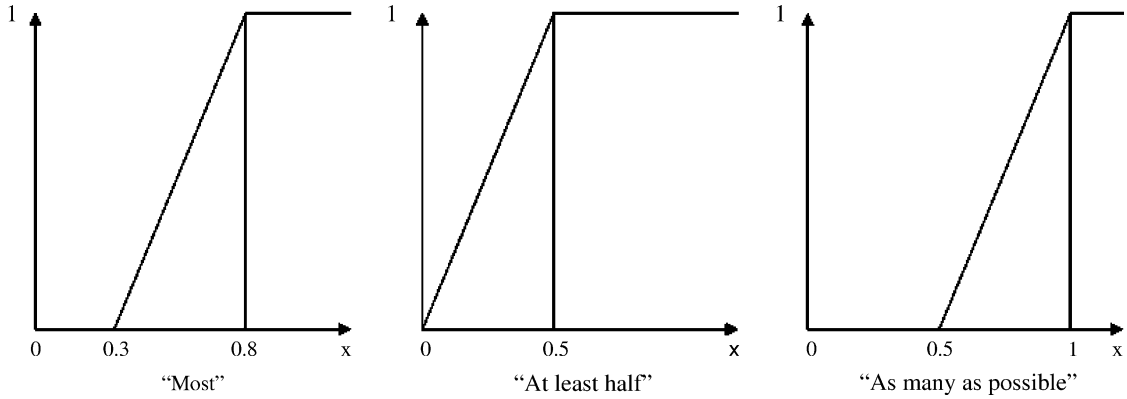

Fig. 2. Proportional fuzzy linguistic quantifiers.

compared to others in a fuzzy majority sense. By again making use of the OWG operator, we have

$$\mathrm{QGID}_i^k = \frac{1}{2}(1 + \log_9 \phi_Q^{\mathrm{G}}(p_{\mathrm{ij}}^k : j = 1, \ldots, N)) \tag{12}$$

for all $i=1,\ldots,N$ (Herrera et al., 2001). Finally, the obtained QGID_i^k values must be normalized, i.e. $\mathrm{QGID}_i^k = \mathrm{QGID}_i^k / \sum_i \mathrm{QGID}_i^k$ to have the importance degrees in percentage for group k. This step must be pursued in all CRs hierarchy levels. The importance degree of each hierarchy leaf node requirement is calculated by multiplying its importance value with its up level requirements' importance values. Finally, we calculate the sum product of each CR's group importance value by the group importance percentage to obtain the aggregate CR importance.

The remaining steps encompass the classical QFD calculations. Once the DRs are determined, one or more cross-functional teams can evaluate the strength of their relationship between CRs. A similar approach to the above group decision making process can also be used here but it is not our main focus in this study. The importance of each DR is then calculated as the sum of each CR importance value multiplied by the quantified relationship between the same CR and the current DR. Then the DRs' implementation order can be determined easily by a simple ranking of these importance values.

4. An illustrative example: developing word processing software

To illustrate our approach, we give an illustrative example on the development of a word processing software. The customer requirements and their respective product features are adapted from the example given in (QFD Capture, 1998).

Three focus groups, namely secretarial, business and home computer users, are selected with 50, 30 and 20% importance, respectively. The CRs hierarchy decided by the development experts is shown in Table 1.

We have asked users to evaluate each CR within allowed formats. As an example, seven home computer users (focus group 3) have evaluated the primary CR r^5 = 'support' ($m=5$) and its four secondary CRs ($N_m=4$), namely r^{51} = 'useful on-line help', r^{52} = 'easy to register', r^{53} = 'provides world-class hotline support' and r^{54} = 'no cost bug fixing', such that

Table 1
Hierarchical customer requirements for word processing software

Before purchase (r^1)		Operates with our network (r^{11})
		Supports the printers I need (r^{12})
		Little or no training required (r^{13})
		Good warranty (r^{14})
		Good price/performance (r^{15})
		Product will stay up-to-date (r^{16})
First impressions (r^2)		Well packaged (r^{21})
		Easy to install (r^{22})
		Looks good (r^{23})
		Looks like our other applications (r^{24})
		Easy to learn initially (r^{25})
		Easy to use after learned (r^{26})
		Helps with grammar, spelling, and punctuation (r^{27})
		Documentation is complete and accurate (r^{28})
		Provides lots of examples we can build on (r^{29})
While operating (r^3)		Accurately presents the format of the document (r^{31})
		Finishes operations quickly (r^{32})
		Finishes operations reliably (r^{33})
		Suggests approaches to common problems (r^{34})
		Small disk space requirements (r^{35})
		Efficiently uses memory (r^{36})
		Promotes creativity (r^{37})
Specific features (r^4)	Data management (r^{41})	Helps efficiently organize document (r^{411})
		Allows us to restart where we stopped (r^{412})
		Automatically saves our data (r^{413})
		Allows me to save versions of my data (r^{414})
		Supports concurrent document development by team (r^{415})
		Allows me to document and highlight changes (r^{416})
		Allows integrated text and graphics (r^{417})
	User interface (r^{42})	Allows user to customize the screen layout (r^{421})
		Usable by individuals with different skill levels (r^{422})
		Supports world-wide operations (r^{423})
		Easy to upgrade capabilities (r^{424})
		Allows user to write and save useful macros (r^{425})
	Others (r^{43})	Directly communicates with related applications (r^{431})
		Reads and Writes in common word processor formats (r^{432})
		Lets user do anything printer will allow (r^{433})
Support (r^5)		Useful on-line help (r^{51})
		Easy to register (r^{52})
		Provides world-class hotline support (r^{53})
		No cost bug fixing (r^{54})

– e^{31} provides an ordered importance vector {1, 2, 4, 3}.
– e^{32} gives an importance degree vector such that {0.8, 0.5, 0.4, 0.7}.
– e^{33} rates CRs in linguistic terms {‘VI’, ‘I’, ‘MI’, ‘I’}.
– e^{34} supplies a subset of CRs {r^{51}, r^{53}, r^{54}} that he found important.

– e^{35} provides a subset of CRs and rates them in linguistic terms $\{r^{51}$: 'VI', r^{54}: 'I'$\}$
– e^{36} finds r^{51} and r^{53} more important than r^{51} and r^{54}, and r^{54} more important than r^{52}.
– e^{37} provides the following pair wise comparison matrix:

$$P^{37} = \begin{bmatrix} 1 & 5 & 2 & 3 \\ 1/5 & 1 & 1/3 & 1/3 \\ 1/2 & 3 & 1 & 1/2 \\ 1/3 & 3 & 2 & 1 \end{bmatrix}.$$

By using Eqs. (1)–(6), importance relation matrices P^{31} to P^{36} are computed as below:

$$P^{31} = \begin{bmatrix} 1 & 9^{1/3} & 9^{1} & 9^{2/3} \\ 9^{-1/3} & 1 & 9^{2/3} & 9^{1/3} \\ 9^{-1} & 9^{-2/3} & 1 & 9^{-1/3} \\ 9^{-2/3} & 9^{-1/3} & 9^{1/3} & 1 \end{bmatrix}, \quad P^{32} = \begin{bmatrix} 1 & 8/5 & 2 & 8/7 \\ 5/8 & 1 & 5/4 & 5/7 \\ 1/2 & 4/5 & 1 & 4/7 \\ 7/8 & 7/5 & 7/4 & 1 \end{bmatrix}$$

$$P^{33} = \begin{bmatrix} 1 & 9^{0.25} & 9^{0.50} & 9^{0.25} \\ 9^{-0.25} & 1 & 9^{0.25} & 1 \\ 9^{-0.50} & 9^{-0.25} & 1 & 9^{-0.25} \\ 9^{-0.25} & 1 & 9^{0.25} & 1 \end{bmatrix} \quad P^{34} = \begin{bmatrix} 1 & 9 & 1 & 1 \\ 1/9 & 1 & 1/9 & 1/9 \\ 1 & 9 & 1 & 1 \\ 1 & 9 & 1 & 1 \end{bmatrix},$$

$$P^{35} = \begin{bmatrix} 1 & 9^{0.50} & 9^{0.50} & 9^{0.25} \\ 9^{-0.50} & 1 & 1 & 9^{-0.25} \\ 9^{-0.50} & 1 & 1 & 9^{-0.25} \\ 9^{-0.25} & 9^{0.25} & 9^{0.25} & 1 \end{bmatrix}, \quad P^{36} = \begin{bmatrix} 1 & 9 & 1 & 9 \\ 1/9 & 1 & 1/9 & 1/9 \\ 1 & 9 & 1 & 9 \\ 1/9 & 9 & 1/9 & 1 \end{bmatrix}$$

Taking into account the matrices P^{31}–P^{37}, the OWG operator with fuzzy linguistic quantifier 'at least half' is used to find the group 3 importance relation matrix with weighting vector being (0.2857, 0.2857, 0.2857, 0.1429, 0, 0, 0):

$$P^{3} = \begin{bmatrix} 1.0000 & 6.5033 & 3.8750 & 4.2149 \\ 0.5182 & 1.0000 & 1.8949 & 1.0352 \\ 0.7430 & 4.8036 & 1.0000 & 1.4804 \\ 0.7606 & 5.1959 & 1.9072 & 1.0000 \end{bmatrix}$$

Eq. (11) is used to compute focus group aggregated importance values with weighting vector (0.00, 0.40, 0.50, 0.10) corresponding to the fuzzy linguistic quantifier 'most'. Then, the collaborative importance values of focus group 3 are calculated as (0.7851, 0.4882, 0.5289, 0.5525) which is then

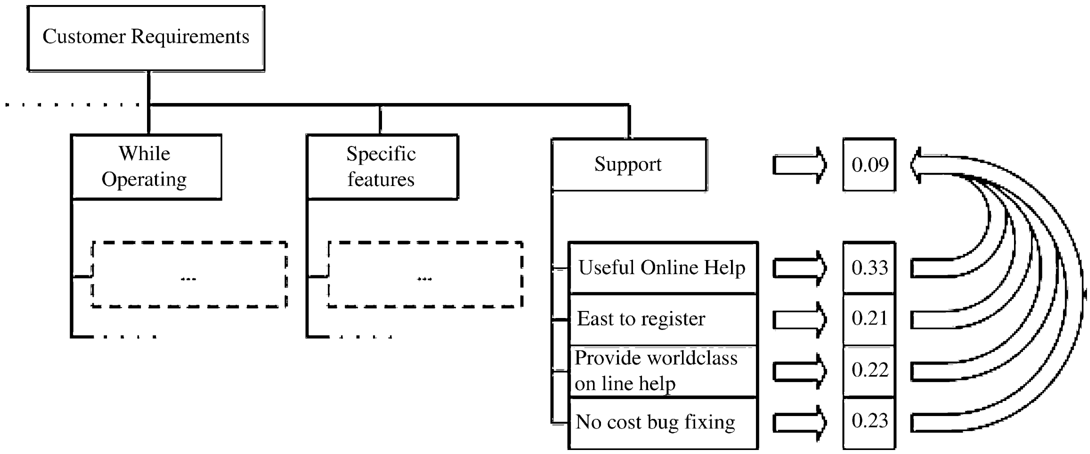

Fig. 3. Finding the importance values of down level requirements.

normalized as (0.33, 0.21, 0.22, 0.23). The relative quantifiers selected here and previously are dependent to the majority concept that is decided by the interviewer. While the quantifier *at least half-* enables us to aggregate highest scores by ignoring lowest values, the quantifier *most* aggregates average scores by canceling mostly highest and lowest values.

Using the same reasoning, hierarchically one level up CR r^5 = 'support' importance is determined as 0.0977. Multiplying this value with each of the secondary CRs r^{51}, r^{52}, r^{53} and r^{54} importance values produces the importance vector of secondary CRs as (0.0326, 0.0203, 0.0219, 0.0229). This process is illustrated in Fig. 3.

Again following the previous steps, we get importance vectors of the same CRs such that (0.0435, 0.0192, 0.0274, 0.0077) and (0.0356, 0.0131, 0.0265, 0.0226) for focus groups 1 and 2, respectively. The overall importance of each CR is then calculated as the weighted sum of the groups' importance values obtained for that CR. Given that focus group weights are given as (0.5, 0.3, 0.2), the final importance value of (r^{51}, r^{52}, r^{53}, r^{54}) is (0.0389, 0.0176, 0.0260, 0.0152) (Fig. 4).

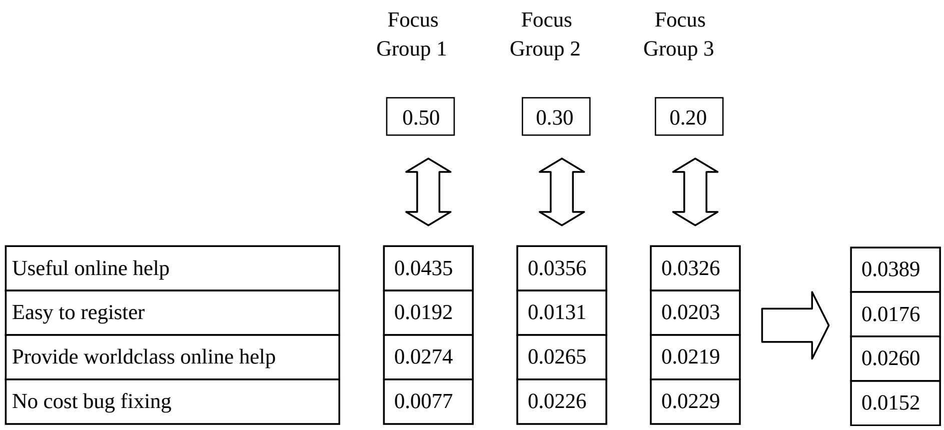

Fig. 4. Aggregating group importance values.

Relationship Strength
Strong : 9
Medium : 3
Weak : 1
None : Blank

		Networks Supported (List)	System Hardware Requirements	Printers Supported (List)	Hours of Training Recommended	Days of "No-Questions" Returns	Suggested Retail Price	Number of Releases Planned Over Next 5 Years	Packaging Layout	Estimated Time for Novice to Install	Preliminary Screen Designs	Number of Deviations From User Interface Guidelines	Estimated Time for Novice to Learn	Interaction Time For Common Operations	Percent of Common Grammar, Spelling, & Punctuation Problems Fo	Preliminary Documentation	Number of Features Included in Sample Documents	Difference of Hard copy From Presentation	Estimated Execution Time to Complete Common Operations	FMEA Rating of Design	Error Processing Messages	Recommended Disk Space (MByte)	Recommended Memory (KByte)	Number of Creativity Tools From List	Number of Means of Organizing Data (List)	Number of Steps Potentially Lost in Power Failure	Frequency Range of Automatic Backup	Number of Versions Allowed	Number of Concurrent Users Allowed	Number of Highlight Methods (List)	Number of Graphic File Formats Displayable	Percent of User Interface Customizable	Number of Operations Configurable to Skill Level	Character Sets Directly Supported (List)	Estimated Time to Change Operations Available to User	Estimated Time to Create Common Macros	Applications Which Can Exchange Data (List)	Word Processor Formats Supported	Percent Printer Capabilities Supported	On-Line Preliminary Help	Estimated Cost ($ & Time) of Registration	Estimated Cost ($ & Time) to Resolve Common Problems	Estimated Cost of Bug Fixes	Business Users (Secretarial) - 0.5	Business Users (Non-Secretarial) - 0.3	Home Computer Users - 0.2	Aggregate Importance
Before Purchase	Operates with our network	9	3	1																									3															1.75%	1.35%	0.91%	1.46%
	Supports the printers I need	1	1	9														1													1								3					2.95%	3.50%	3.35%	3.19%
	Little or no training required				9					1	1	3	3			1	1				1												1							1		3		3.92%	2.42%	1.22%	2.93%
	Good warranty					9		1																																			3	1.03%	0.69%	2.24%	1.17%
	Good price/performance		3				9		3																				3															1.15%	1.53%	2.81%	1.59%
	Product will stay up-to-date							9																																				1.37%	2.68%	1.64%	1.82%
First Impressions	Well packaged						1		9	1						1																									1	1		2.07%	0.85%	3.38%	1.97%
	Easy to install		1			1				9						3																1	1		1					1		1		1.10%	1.83%	3.20%	1.74%
	Looks good								3		9	3				1		1														1								1				1.82%	3.50%	2.38%	2.44%
	Looks like our other applications										3	9	1								1																							2.49%	2.50%	1.57%	2.31%
	Easy to learn initially				9						1	3	9			3	3																							3		1		1.99%	2.97%	2.25%	2.34%
	Easy to use after learned				1	1					3	3		9		3	1		3		1												1									1		5.27%	1.72%	2.18%	3.59%
	Helps with grammar, spelling, and punctuation													1	9	1			1		1											3	1			1								5.03%	4.21%	2.14%	4.21%
	Documentation is complete and accurate				1					1			3			9																								3		1		3.83%	2.86%	2.93%	3.36%
	Provides lots of examples we can build on				1								1			3	9																											1.26%	4.40%	4.82%	2.91%
While Operating	Accurately presents the document			3							1	1						9													3								3					4.65%	2.03%	3.86%	3.71%
	Finishes operations quickly																		9				3																					4.07%	2.78%	1.55%	3.18%
	Finishes operations reliably																		1	9	1																							4.55%	3.08%	3.14%	3.82%
	Suggests approaches to common problems												1	1	3	1	3				9																			3		1		1.99%	1.51%	1.86%	1.82%
	Small disk space requirements		1																3			9	3					1																0.89%	2.33%	2.30%	1.61%
	Efficiently uses memory		1											1					1				9																					1.01%	1.76%	3.00%	1.64%
	Promotes creativity										1			3		1	1		3					9	1												1		1					0.76%	4.44%	2.22%	2.16%
Specific Features – Data Management	Helps efficiently organize document														1	1	1							3	9				1		1									1				0.31%	3.26%	3.61%	1.85%
	Allows us to restart where we stopped																									9																		4.36%	2.92%	3.46%	3.74%
	Automatically saves our data (Option)																									3	9	1																2.80%	1.65%	3.01%	2.50%
	Allows me to save versions of my data																					3				1	1	9																1.13%	2.99%	1.01%	1.67%
	Supports concurrent document development by team	3	1		1											1	3							1					9	3										1				1.29%	1.86%	1.36%	1.48%
	Allows me to document & highlight changes			1											1			1												9														4.57%	3.06%	0.69%	3.34%
	Allows integrated text and graphics		3	1													3	3						1						1	9								1					3.40%	2.13%	4.72%	3.28%
Specific Features – User Interface	Allows user to customize the screen layout																															9	3		1	3								1.50%	1.25%	2.02%	1.53%
	Usable by individuals with different skill levels																																9		1									1.88%	3.17%	1.49%	2.19%
	Supports world-wide operations														3																			9						1				1.35%	0.65%	1.11%	1.09%
	Easy to upgrade capabilities																															1	1		9									1.33%	2.50%	1.58%	1.73%
	Allows user to write and save useful macros													3																						9								2.68%	1.19%	2.55%	2.21%
Specific Features – Others	Directly communicates with related applications																																				9	3						2.51%	2.19%	3.43%	2.60%
	Reads & Writes in common word processor formats																																				3	9						2.84%	2.74%	1.73%	2.59%
	Lets user do anything printer will allow			3														3																1					9					3.32%	3.74%	3.50%	3.48%
Support	Useful on-line help																																							9		1		4.35%	3.56%	3.26%	3.89%
	Easy to register								1																																9			1.92%	1.31%	2.03%	1.76%
	Provides world-class hotline support																																							1		9		2.74%	2.65%	2.19%	2.60%
	No cost bug fixing							1																																			9	0.77%	2.26%	2.29%	1.52%
	Importance of product characteristics	0.21	0.29	0.58	0.59	0.16	0.16	0.19	0.32	0.24	0.51	0.58	0.47	0.53	0.52	0.81	0.63	0.63	0.60	0.34	0.33	0.19	0.29	0.30	0.19	0.43	0.24	0.19	0.24	0.38	0.46	0.32	0.39	0.13	0.21	0.29	0.33	0.31	0.57	0.72	0.18	0.51	0.17				
	Importance of product characteristics (%)	1.3%	1.8%	3.7%	3.7%	1.0%	1.0%	1.2%	2.0%	1.5%	3.2%	3.7%	3.0%	3.4%	3.3%	5.1%	4.0%	4.0%	3.8%	2.2%	2.1%	1.2%	1.8%	1.9%	1.2%	2.7%	1.5%	1.2%	1.5%	2.4%	2.9%	2.1%	2.4%	0.8%	1.3%	1.8%	2.1%	2.0%	3.7%	4.6%	1.1%	3.2%	1.1%				
	Ranking	33	27	7	6	41	40	36	23	31	13	8	14	10	11	1	3	4	5	19	21	34	26	25	37	16	30	35	29	18	15	22	17	42	32	28	20	24	9	2	38	12	39				

Fig. 5. The final HOQ for word processing software.

Table 2
Ten most important technical requirements for word processing software

Functional requirements	Importance	% Importance	Ranking
Preliminary documentation	0.8078	5.14	1
On-line preliminary help	0.7168	4.56	2
Number of features included in sample documents	0.6348	4.04	3
Difference of hardcopy from presentation	0.6261	3.98	4
Estimated execution time to complete common operations	0.6034	3.84	5
Hours of training recommended	0.5875	3.73	6
Printers supported	0.5838	3.71	7
Number of deviations from user interface guidelines	0.5835	3.71	8
Percent printer capabilities supported	0.5747	3.65	9
Interaction time for common operations	0.5302	3.37	10

When all CR importance values are computed and relationship strengths with technical characteristic are determined (see Fig. 5), the importance of control characteristics can be found as explained in Section 3. Table 2 shows the 10 most important technical requirements detected.

5. Concluding remarks and extensions

As a customer-driven quality management system, QFD involves numerous input data from both customers and QFD team members. Depending on their background, people give information about their personal preferences in many different ways. As the determination of CR priorities is the key concept in QFD, we believe that greater emphasis has to be given to analyze and merge individual assessments in different formats. In this paper, we presented an approach that enables to combine linguistic and numerical information for QFD and determined the CRs' importance values based on the fuzzy majority concept. While the approach is applied to software development, it is worthwhile to note that it can also be applied to different industries.

Though not difficult to include, the correlation between DRs is not considered in this study to keep the focus on the proposed group decision-making approach. This may be a subject of future research. Since we treated focus group members equally, another concern can be to introduce a hierarchy/ranking between members before detecting collaborative CR importance values. This can be accomplished by using importance induced ordered weighted geometric (I-IOWG) operator as given in Chiclana, Herrera-Viedma, Herrera, and Alonso (2004) and Herrera, Herrera-Viedma, and Chiclana (2003). It would be beneficial also to extend the proposed method to subsequent HOQs.

References

Akao, Y. (1972). New product development and quality assurance deployment system. *Standardization and Quality Control*, *25*(4), 243–246 (in Japanese).

Akao, Y. (Ed.). (1990). *QFD—Quality function deployment: integrating customer requirements into product design*. Portland: Productivity Press.

Basili, V. R., & Musa, J. D. (1991). The future engineering of software—A management perspective. *Computer*, *24*(9), 90–96.

Bouchereau, V., & Rowlands, H. (2000). Methods and techniques help quality function deployment (QFD). *Benchmarking: An International Journal*, *7*(1), 8–20.

Chan, L. K., & Wu, M. L. (2002). Quality function deployment: A literature review. *European Journal of Operational Research*, *143*(3), 463–497.

Chiclana, F., Herrera, F., & Herrera-Viedma, E. (1998). Integrating three representation models in fuzzy multipurpose decision making based on fuzzy preference relations. *Fuzzy Sets and Systems*, *97*, 33–48.

Chiclana, F., Herrera-Viedma, E., Herrera, F., & Alonso, S. (2004). Induced ordered weighted geometric operators and their use in the aggregation of multiplicative preference relations. *International Journal of Intelligent Systems*, *19*, 233–255.

Cox, C. A. (1992). Keys to success in quality function deployment. *APICS—The Performance Advantage* , 25–28.

Davis, A. M. (1993). *Software requirements: Objects, functions and states*. Englewood Cliffs, NJ: Prentice-Hall.

Dromey, R. G. (2003). Software quality—Prevention versus cure? *Software Quality Journal*, *11*, 197–210.

Dubois, D., & Prade, H. (2004). On the use of aggregation operations in information fusion processes. *Fuzzy Sets and Systems*, *142*(1), 143–161.

Dunn, R. H. (1988). Software quality assurance: A management perspective. *Quality Progress*, *21*(7), 52–56.

Elboushi, M. I., & Sherif, J. S. (1997). Object-oriented software design utilizing quality function deployment. *Journal of Systems and Software*, *38*(2), 133–143.

Erol, I., & Ferrell, W. G. (2003). A methodology for selection problems with multiple, conflicting objectives and both qualitative and quantitative criteria. *International Journal of Production Economics*, *86*(3), 187–199.

Gong, B., Yen, B. Y., & Chou, D. C. (1998). A manager's guide to total quality design. *Industrial Management and Data Systems*, *8*, 100–107.

Govers, P. M. (2001). QFD not just a tool but a way of quality management. *International Journal of Production Economics*, *69*, 151–159.

Haag, S. E. (1992). *A field study of the use of quality function deployment as applied to software development*. PhD Thesis. Arlington: University of Texas.

Haag, S. E. (1993). *A quality paradigm for software engineering Seminar notes*. Arlington: University of Texas.

Haag, S., Raja, M. K., & Schkade, L. L. (1996). Quality function deployment usage in software development. *Communications of the ACM*, *39*(1), 41–49.

Hales, R. (1995). Adapting QFD to the US. *IIE Solutions*, *27*, 15–18.

Han, S. B., Chen, S. K., Ebrahimpour, M., & Sodhi, M. S. (2001). A conceptual QFD planning model. *International Journal of Quality and Reliability Management*, *18*(8), 796–812.

Harding, J. A., Popplewell, K., Fung, R. Y. K., & Omar, A. R. (2001). An intelligent information framework relating customer requirements and product characteristics. *Computers in Industry*, *44*(1), 51–65.

Herrera, F., Herrera-Viedma, E., & Chiclana, F. (2001). Multiperson decision-making based on multiplicative preference relations. *European Journal of Operational Research*, *129*, 372–385.

Herrera, F., Herrera-Viedma, E., & Chiclana, F. (2003). A study of the origin and uses of the ordered weighted geometric operator in multicriteria decision making. *International Journal of Intelligent Systems*, *18*, 689–707.

Herzwurm, G., & Schockert, S. (2003). The leading edge in QFD for software and electronic Business. *International Journal of Quality and Reliability Management*, *20*(1), 36–55.

Herzwurm, G., Schockert, S., & Mellis, W. (2000). *Joint requirements engineering. QFD for rapid customer-focused software and internet development*. Braunschweig/Wiesbaden: Vieweg/Gabler.

Ho, E. S. S. A., Lai, Y. J., & Chang, S. I. (1999). An integrated group decision-making approach to quality function deployment. *IIE Transactions*, *31*, 553–567.

Issac, G., Rajendran, C., & Anantharaman, R. N. (2004). A conceptual framework for total quality management in software organizations. *Total Quality Management*, *15*(3), 307–344.

Karlsson, J. (1997). Managing software requirements using quality function deployment. *Software Quality Journal*, *6*, 311–325.

Kudikyala, U. K., & Vaughn, R. B. (2005). Software requirement understanding using Pathfinder networks: Discovering and evaluating mental models. *The Journal of Systems and Software. 74*, 101–108.

Lai, Y. J., Ho, E. S. S. A., & Chang, S. I. (1998). Identifying customer preferences in quality function deployment using group decision-making techniques. In J. M. Usher, U. Roy, & H. R. Parsaei (Eds.), *Integrated product and process development: Methods, tools, and technologies* (pp. 1–28). New York: Wiley, 1–28 (Chapter 1).

Lee, J. W., & Kim, S. H. (2000). Using analytic network process and goal programming for interdependent information system project selection. *Computers & Operations Research*, *27*, 367–382.

Liu, X. F. (1998). A quantitative approach for assessing the priorities of software quality requirements. *The Journal of Systems and Software*, *42*, 105–113.

Liu, X. F. (2001). Software quality function deployment. *IEEE Potentials*, *19*(5), 14–16.

Liu, X. F., Jia, R. Q., & Viswanathan, R. (1998). An intelligent tool for analysis of imprecise software quality requirements from different perspectives. *Concurrent Engineering—Research and Applications*, *6*(3), 207–223.

Ma, M., Kandel, A., & Friedman, M. (2000). A new approach for defuzzification. *Fuzzy Sets and Systems*, *111*, 351–356.

Masud, A. S. M., & Dean, E. B. (1993). Using fuzzy sets in quality function deployment. In: *Proceedings of the second industrial engineering research conference, Los Angeles, CA*, May 26–27.

O'Brien, D. H. (1991). Software quality starts with the customer. *Quality Progress*, *30*(6), 22–24.

QFD Capture (1998). Milford, USA: International TechneGroup Inc.

Richardson, I. (2001). Software process matrix: A small company SPI model. *Software Process: Improvement and Practice*, *6*(3), 157–165.

Saaty, T. L. (1980). *The analytic hierarchy process*. New York: McGraw Hill.

Sivaloganathan, S., & Evbuomwan, N. F. O. (1997). Quality function deployment-the technique: State of the art and future directions. *Concurrent Engineering, Research and Applications*, *5*(2), 171–180.

Verma, D., Smith, C., & Fabrycky, W. (1999). Fuzzy set based multi-attribute conceptual design evaluation. *Systems Engineering*, *2*(4), 187–197.

Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decision-making. *IEEE Transactions on Systems, Man and Cybernetics*, *18*(1), 183–190.

Wallace, T. F. (1992). *Customer driven strategy: Winning through operational excellence*. Vermont, MD: Oliver Wight Publications, Inc.

Xu, Z. (2004). A method based on linguistic aggregation operators for group decision making with linguistic preference relations. *Information Sciences. 166*, 19–30.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*, 338–353.

Zadeh, L. A. (1983). A computational approach to fuzzy quantifiers in natural languages. *Computers & Mathematics with Applications*, *9*, 149–184.

Zhang, Q, Chena, J. C. H., & Chong, P. P. (2004). Decision consolidation: Criteria weight determination using multiple preference formats. *Decision Support Systems. 38*, 247–258.

Zhou, M. (1998). Fuzzy logic and optimization models for implementing QFD. *Computers and Industrial Engineering*, *35*(1-2), 237–240.

Zultner, R. E. (1992). Quality function deployment (QFD) for software: Structured requirements exploration. In G. G. Schulmeyer, & J. I. McManus (Eds.), *Total quality management for software* (pp. 297–319). New York: Van Nostrand Reinhold, 297–319 (Chapter 12).