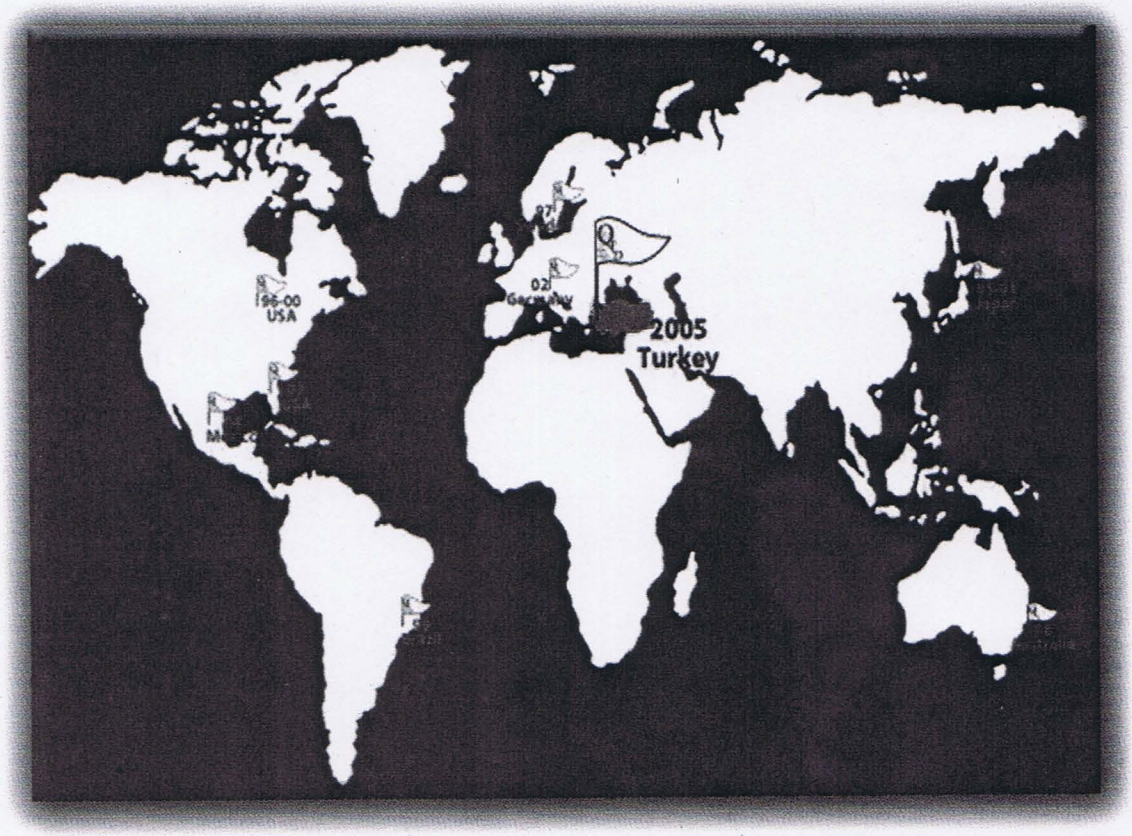
International Council for Quality Function Deployment

DOKUZ EYLÜL UNIVERSITY
FACULTY OF BUSINESS

PROCEEDINGS



11th INTERNATIONAL SYMPOSIUM ON QUALITY FUNCTION DEPLOYMENT

26-30th September 2005

Kuşadası, TURKEY

INDEX

KEYNOTE SPEECHES

QFD & THEORY

INFORMATION TECHNOLOGY

PRODUCT & PROCESS DEVELOPMENT

A Software Tool For Making A Two-Way Table Like A Quality Table

Yunarso Anang
SYNC Information Systems, Co. Ltd., JAPAN, E-mail: y-anang@syncinfo.co.jp; Working doctoral student at Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi

Akio Amemiya
Master's course student, ibid. E-mail: akio_a@is.esi.yamanashi.ac.jp

Masanobu Yoshikawa
Research Associate, Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi, Address: 4-3-11 Takeda, Kofu, Yamanashi 400-8511 JAPAN, PHONE: +81-55-220-8491, FAX: +81-55-220-8491, E-mail: yoshi@yamanashi.ac.jp

Yoshimichi Watanabe
Associate Professor, ibid, PHONE: +81-55-220-8651, FAX: +81-55-220-8651, E-mail: nabe@yamanashi.ac.jp

Hisakazu Shindo
Professor, ibid, PHONE: +81-55-220-8499, E-mail: h-shindo@yamanashi.ac.jp

Abstract

Since Quality Function Deployment (QFD) was proposed by Yoji Akao, QFD researchers and practitioners could not help but accept a considerable effort to make quality tables. However it seems to be true that some would have refused the effort because of its amount. Even if QFD is effective to develop a new product or service, it should be resolved that QFD requires considerable amount of work for making tables.

So far, MS Excel has been used to make tables which are required for QFD. It results from a fact that MS Excel was designed for manipulating tables. However, we were not able to be satisfied with it. The reason is that we make a table by using another means such as the "KJ method" or the "Affinity Diagram Method" and then input it into computer through MS Excel. This means that a tool which helps not only the process of input but also the process of making tables would be necessary.

In this report we propose a new software tool that directly helps one make tables such as a quality table. Moreover, we are able to make a table well-structured by using the Quantification Method of type three (QM3). The resultant table is easily transferred to MS Excel through a clipboard function. The development is being carried out from the point of view of the fusion of UML and QFD.

1. Introduction

Since Quality Function Deployment (QFD) was proposed by Yoji Akao, QFD researchers and practitioners could not help accepting a large amount of effort to make various tables [1,2]. However it seems to be true that some would not have accepted the effort of making tables because of its much work. Even if QFD is effective to develop a new product or service, it should be resolved that QFD requires considerable amount of work for making tables [3].

So far, MS Excel has been used to make tables which are required for QFD [4]. It results from the fact that MS Excel was designed for manipulating tables. However, we were not able to be satisfied with it. The reason is that we make a table by using another means such as the "KJ method" or the "Affinity Diagram Method" and then input it into computer through MS Excel. The development process can be made more efficient by integration with idea generation support. This means that a tool which helps not only the process of input but also the process of making tables would be necessary.

In this report we propose a new software tool that directly helps one make tables such as quality tables. The associative listing of items can be utilized through making a table. Moreover, we are able to make a table well-structured by using the Quantification Method of type three (QM3). Because of different ways of expression or other reasons, the table in the early stage of development may contain similar items. This tool helps subgrouping of such items to be merged into one.

The resultant table is easily transferred to MS Excel through a clipboard function. The development is being carried out from the point of view of the fusion of UML and QFD.

2. A framework of an application of QFD to developing software tool

From the point of view of quality assurance, a series of procedure of QFD is thought to have been almost established in developing a tangible product. However, applying QFD to software development remains unsatisfactory. In software engineering field, a recent attention is being headed for UML (Unified Modeling Language). Reflecting such trend of the software engineering, Watanabe et al. [5] proposed a fusion of QFD and UML to develop a software system. We are conscious of their proposition and set a framework to apply QFD to developing a software tool.

Following the input-process-output model, the framework is depicted as shown in Figure 1. The most important thing is, needless to say, to understand use cases employed in UML. We can then derive the information regarding how to use and requirements for quality. These should be considered in designing the tool. As output we need to determine component layouts on each screens and menus. The process can be regarded as methods or algorithms that transform input data into output data.

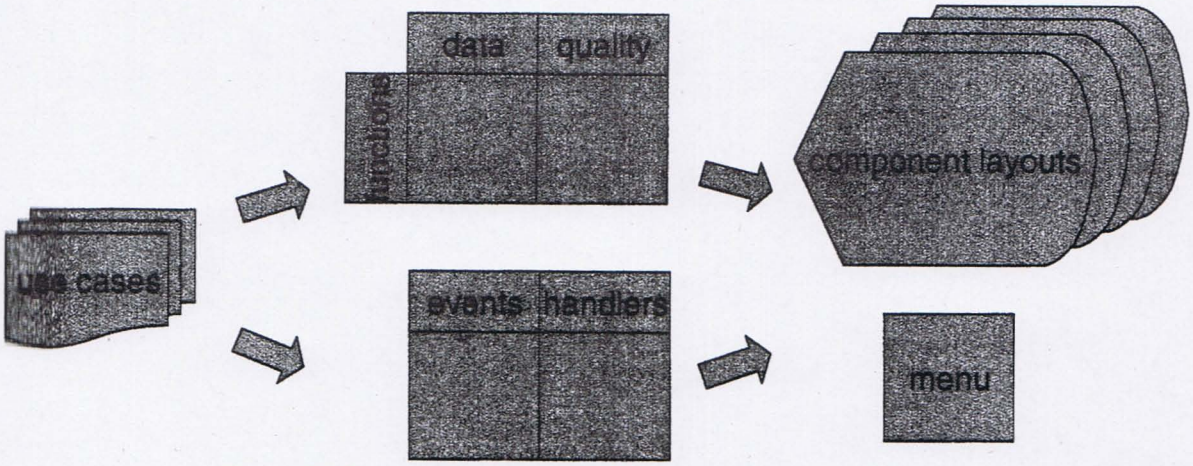


Figure 1 A framework of applying QFD to developing the software tool

3. Enforcement of the framework

3.1 Gathering use cases

In order to grasp use cases, we first examined traditional procedures of QFD and those used in existing QFD tools. However users themselves can not always recognize all the requirements for the tool. Besides, it is difficult for users to express adequate requirements because the tool has not been clarified and completed yet. Therefore, it is of practical importance for us to try to derive latent requirements for the tool to be realized.

From the point of view of developing the new tool, it should provide new functions or improvements. Although there have been many software tools for QFD, it can be said that no tool has helped us put every procedures of QFD into practice. One of the reasons is that tools have left most burdensome and difficult works to practitioners. In fact, these works have been almost embedded in "KJ method" or the "Affinity Diagram Method".

From above, we determined the following ideas of the tool to be developed:

(1) We can optionally pick out two viewpoints which may constitute an object of QFD such as "quality requirements" and "quality elements".

(2) Input is done in order of users' hitting on the idea of items of each viewpoint.

(3) Items of each viewpoint can be input in such a manner as "A1" relates "B1" with degrees of relationship 1-9 or "?". The input symbol "?" is temporarily used in the case that it is difficult to determine the value of relationship between items. Needless to say, the value should be determined and input as a definite value later on.

(4) A two-way table is automatically composed on the basis of input information at any time users like. If the table contains temporary values, it is indicated in order to be substituted by their definite values.

(5) "Quantification Method of Type Three (QM3)" should be applied to the two-way table instead of "KJ Method" or "Affinity Diagram Method". The QM3 will give us necessary information for subgrouping items of each viewpoint. It is important that subgrouping is simultaneously carried out against two viewpoints.

(6) Tree-view components enable users to manipulate hierarchies of items of each viewpoint.

(7) The finally established two-way table can be exported to other application program such as MS Excel.

3.2 Functions and data derived from use cases

We examined use cases and derived relevant functions and data as shown in Table 1. In this step, it is important to be aware of sequences of the QFD process.

Table 1 Functions and data derived from use cases

No.	Use case	Function	Data
1	Input a two-way table	Input two viewpoints	Viewpoints A and B
		Input items of each viewpoint and its degree of relationship	Item of A, Item of B, and its degree of relationship
2	View a table	Display a two-way table	Ditto
		Display each items by tree-view	Ditto
3	Edit items and hierarchy	Edit items	Table, item, degree
		Edit hierarchy	Tree-views
4	Analyze the table	Apply QM3 to the table	Table, eigen values, and scores of each items
5	Prioritize items	Compute weights of items	Weights of "A" and "B" items
6	Reuse a table	Import a table	Table data
		Export a table	Table data

Table 2 A table for designing the menu

File	Create	Specify two viewpoints; input items and degrees of relationship between them
	Save	Overwrite data on the already opened file
	Save as	Input the name of the file and add the extension code
	Open	Specify the name of the file
	Import	Specify the name of the csv format file
	Export	Specify the name of the file
	Print	Specify the name of data
	Quit	Complete all processing
Edit	Initialize	Clear the tree-view components
	Add	Input the item
	Add Child	Input the item and specify the relevant item and its degrees of relationship
	Merge	Specify items to be merged
	Delete	Specify item (s) to be deleted
	Delete Children	Specify the item
Display	Two-way Table	Display the data in a table form
	Tree-view	Display the data in a tree-view form
Tool	QM3	Sort the table with respect to specified scores
		Display the scatter diagram with specified eigen vectors
	Prioritization	Convert weights
Help	Version	Display the version information
	Usages	Display the usages

3. 3 Design of menus

In order to design the menus of this tool, we extracted functions to be implemented considering above-mentioned ideas and use-cases as shown in Table 2. The menu items basically followed those of MS windows applications.

3.4 Brief specifications of some new functions

The object-oriented system development helps us think about details of implementation. For example, saving data as a file is easily implemented by using the standardized "file save dialogbox". Then we introduce the following three functions that characterize this tool.

(1) A function of creating a two-way table

So far, "KJ method" or "Affinity Diagram Method" has been indispensable to put QFD into practice. In fact, carrying out "KJ Method" twice seems to be necessary for us to make quality tables. It is no exaggeration to say that most of time required for QFD is devoted to making quality tables.

Instead of "KJ Method" or "Affinity Diagram Method", we have proposed the application of "Quantification Method of type three (QM3)" [6]. Of four quantification methods, QM3 is used for analyzing inside structure of two-way tables. Subgrouping items which seem to be similar to each other can be performed on the basis of scores obtained by solving an eigen-value problem in QM3.

Therefore, the first thing we should do is to make a two-way table even if it is not well-formed. The following two ways of inputting items and degrees of relationship were designed for

the tool. At first, we specify two viewpoints "A" and "B". After inputting the item "A1" which belongs to "A", we can input items "B1", "B2", etc. with each degrees of relationship as shown in Figure 3.

Another way of inputting data was designed for inputting items "A1" and "B1" together with degrees of relationship between them. Figure 4 indicates this input method called "the triplet input method".

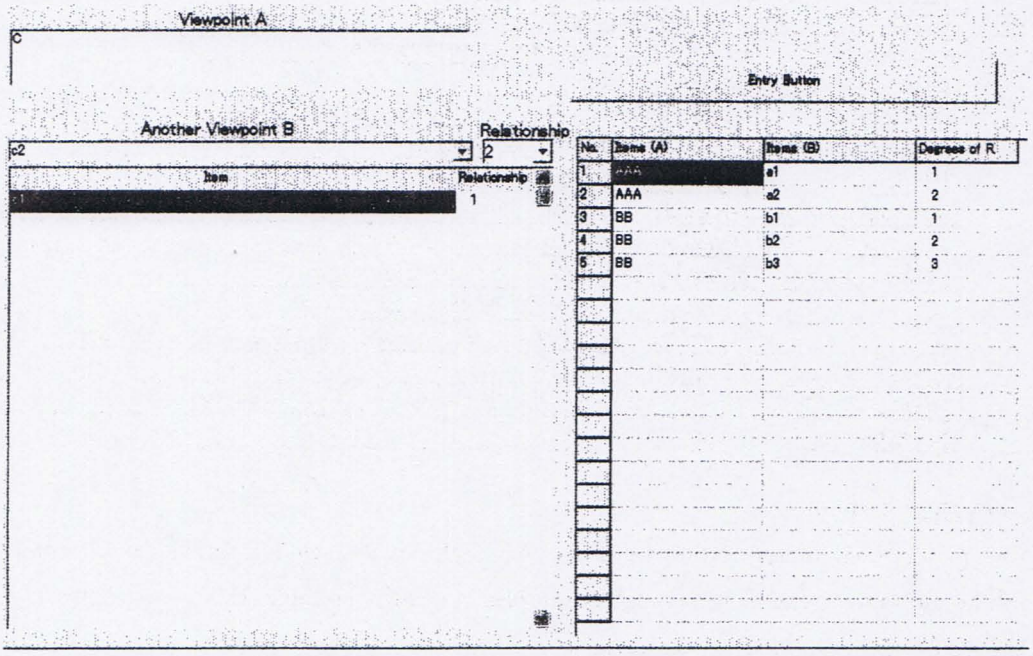


Figure 3 A screen image of inputting items and relationship

No.	Items (A)	Items (B)	Degrees of R
1	AAA	a1	1
2	AAA	a2	2
3	BB	b1	1
4	BB	b2	2
5	BB	b3	3
6	C	c1	1
7	C	c2	2

Figure 4 Triplet input method

(2) Applying QM3

We have already reported QM3 to analyze the inside structure of two-way tables [7]. After completing inputting items and degrees of relationship, the two-way table is automatically composed. This table may contain such similar items as are merged into one. Besides, the table may be missing important items. If so, the sorted table with respect to items' scores provides us with chances of merging or deleting items. Needless to say, the scatter diagram of scores with respect to specified two engen-values helps us examine to merge or delete items.

Table 3 A two-way table

	B1	B2	B3	B4	B5	B6	B7
A1	o			o			
A2					o	o	
A3		o	o				
A4		o	o				
A5	o			o			
A6						o	o

Table 4 Eigen-values and scores

EVNo	1	2	3	4
EV	1.00	1.00	1.00	0.50
A1	1.73	0.00	0.00	0.00
A2	0.00	0.00	1.73	1.73
A3	0.00	1.73	0.00	0.00
A4	0.00	1.73	0.00	0.00
A5	1.73	0.00	0.00	0.00
A6	0.00	0.00	1.73	-1.73
B1	1.73	0.00	0.00	0.00
B2	0.00	1.73	0.00	0.00
B3	0.00	1.73	0.00	0.00
B4	1.73	0.00	0.00	0.00
B5	0.00	0.00	1.73	2.45
B6	0.00	0.00	1.73	0.00
B7	0.00	0.00	1.73	-2.45

Table 5 The sorted table

	B1	B4	B2	B3	B5	B6	B7
A1	o	o					
A5	o	o					
A3			o	o			
A4			o	o			
A2					o	o	
A6						o	o

For example, suppose that a two-way table as shown in Table 3 is given. Applying QM3 to the table, we can get the eigen-values and scores for respective viewpoints. It is possible to make a scatter diagram from these scores, which may bring about useful information to make the table well-structured. Items being sorted with respect to scores belonging to eigen-value number 2, we can get the structuralized table as shown in Table 5. In the case of making a quality table in the traditional QFD, we must use twice the "KJ Method" or the "Affinity Diagram Method". If we resort to QM3 instead of these methods, we can simultaneously sort items by their scores.

(3) Manipulation of hierarchy of items

One of features of QFD is to process a variety of information by hierarchically stracturalizing. Nevertheless, it seems that there have been no existing tools for QFD that support hierarchical processing of information, which results from the technological difficulty of hierarchical information processing.

We then devised a method using "tree-view" component that is widely used in MS windows software applications such as MS "Explorer". This is because it is difficult to edit the hierarchy of items that are in the form of a table. As a consequence, we first edit the hierarchy by the same operations as are employed in MS "Explorer", then we map it into corresponding two-way table. The resultant table is illustrated as shown in Table 6.

Table 6 The structuralized table

		□□□		□□□		□□□□		
		B1	B4	B2	B3	B5	B6	B7
□□□	A1	o	o					
	A5	o	o					
□□□	A3			o	o			
	A4			o	o			
□□□	A2					o	o	
	A6						o	o

4. Menu, events and screen layout

In order to implement these functions so that they are satisfied with users' requirements, we must select suitable software components with considering their layout on each screen. The table 7 shows suitable software components we have selected.

Table 7 Menu, events and screen layout

			screens (windows)				
		Required Components for implementing	1	2	3	4	5
File	Create	Label, ComboBox, F1Book, StringGrid, Button	o	o			
	Save	SaveDialog	o		o		
	Save as	SaveDialog	o		o		
	Open	OpenDialog	o				
	Import	F1Book	o				
	Export	F1Book	o				
	Print	PrintDialog	o				
	Quit	MessageDialog	o				
Edit	Initialize	TreeView, TreeViewPopUpMenu,	o		o		
	Add	TreeView, TreeViewPopUpMenu,	o		o		
	Add Child	TreeView, TreeViewPopUpMenu,	o		o		
	Merge	TreeView, TreeViewPopUpMenu,	o		o		
	Delete	TreeView, TreeViewPopUpMenu,	o		o		
	Delete Children	TreeView, TreeViewPopUpMenu,	o		o		
Display	Two-way Table	F1Book	o	o			
	Tree-view	TreeView, TreeViewPopUpMenu,	o		o		
Tool	QM3	Canvas, TeeChart, F1Book,	o			o	o
		Canvas, TeeChart, F1Book,	o			o	o
	Prioritization	unimplemented	o				
Help	Version	ShowMessage	o				
	Usages	unimplemented	o				

5. Implementation and evaluation

As a software integrated development environment, "Delphi Version 7"[i] was determined. This was because we have been familiar with "Delphi" for the object oriented software development [7,8,9]. The software tool was almost implemented on the basis of the framework that has been proposed as a fusional methodology of QFD and UML. Although the framework should be more examined to establish a fusional methodology, our experience of applying it to developing the tool seemed effective.

The improvement of the tool is now on-going. We will make every effort to make the tool more useful.

Acknowledgement

The authors' gratitude should go to Mr. Naofumi Takayama, the president of Sync Information Systems Co. Ltd. for his suggestions and supports. The authors would like to thank Professor Xiong Wei for his helpful suggestions. The authors also express our acknowledgements to Mr. Shigeru Yamashita for his supports and encouragements.

i. Delphi is a registered trademark of Borland Software Corporation.

References

[1] H. Shindo ed.: Activation of Practical QFD - Creation of new value, JUSE press, 1998 (in Japanese).

[2] H. Shindo ed.: A design-oriented problem solving - An approach to TQM activation, JUSE press, 2001 (in Japanese).

[3] W. Xiong, Y. Watanabe, M. Yoshikawa, and H. Shindo: A Method of Quantitative Software Requirements Analysis and Its Mapping Based on QFD, The 10th International Symposium on Quality Function Deployment, pp. 219-226, 2004.

[4] G. Herzwurm, S. Reis, and S. Schockert: The support of qualiry function development by the customer-oriented evaluation of software tools, The 9th International Symposium on Quality Function Deployment, pp. 139-157, 2003.

[5] Y. Watanabe, Yunarso Anang, M. Yoshikawa, Y. Kasai, and H. Shindo: A new approach to software design with QFD-like tabulations of UML diagrams, The 10th International Symposium on Quality Function Deployment, pp. 193-206, 2004.

[6] W. Xiong, and H. Shindo: An application of quality table concept to the analysis of software structure, The 1st International Symposium on Quality Function Deployment, 1995.

[7] Yunarso Anang, S. Yamashita, Y. Watanabe, and H. Shindo: A Software Tool for Quality Function Deployment Incorporating the Quantification Method of Type 3, The 4th

International Symposium on Quality Function Deployment, pp. 77-84, 1998.

[8] Yunarso Anang, A. Shimizu, A. Kitamura, D. Ozawa, M. Yoshikawa, Y. Watanabe, S. Yamashita, and H. Shindo: Development of a Web-based QFD Tool, The 8th International Symposium on Quality Function Deployment, pp. 371-376, 2002.

[9] M. Yoshikawa, Y. Watanabe, H. Shindo, M. Takeshita, and N. Takayama: e-Learning System for QFD with improving contents and mentoring in "e-TQM" Project, The 9th International Symposium on Quality Function Deployment, pp. 159-167, 2003.