

## A FUTURISTIC SOFTWARE FRAMEWORK TO GENERATE ACTUAL CUSTOMER NEEDS FOR QUALITY FUNCTION DEPLOYMENT

Ashish K. Sharma, I. C. Mehta, Dr. J. R. Sharma,  
Sangita Sharma

---

### ABSTRACT

Quality Function Deployment (QFD) is a product development process that encompasses a sheer amount of data gathered from customers through several market research techniques like personal interview, focus groups, surveys, video conferencing etc. This massive, unsorted and unstructured data is required to be transformed into a limited number of structured information to represent the actual 'Customer Needs'. However the process is tedious and time consuming and cannot be dealt with manually. In order to address these issues, this paper proposes a futuristic software framework based on an Affinity Process. The paper begins with the topic introduction and outlines the QFD process. The paper then describes the Affinity Process, builds the data structure and then makes an attempt to build the proposed framework using tools Visual Basic (VB) and MS-Access. The proposed framework is developed as a part of QFD software and it is anticipated that when completely developed, it would act as a vital component of QFD software.

**Keywords:** QFD, Affinity Process, Visual Basic, MS-Access, Software, Customer Needs.

---

### 1. INTRODUCTION

Quality Function Deployment (QFD) is a customer oriented product development process that is driven by the 'voice of customers'. QFD is a systematic process for helping a business to focus on its priorities, investments and its customers. It uses cross-functional teams to identify and resolve issues involved in providing products, processes, services and strategies to meet or exceed customer expectations (King, 1989). QFD is a customer focused product development process through which the quality of the product can be enormously improved. Companies in practice of utilizing QFD are incessantly seeking for efficient QFD software that could facilitate designers and engineers to evaluate company's decision-making process. QFD comprises of several matrices, and is rather a lengthy process, which involves voluminous data and complex calculations. As a result, dealing with the data manually is quite cumbersome and time consuming (Sharma, Mehta and Sharma, 2009). In QFD, the first key to the development of high quality products is to understand the customer needs thoroughly. Since customers are integral part of QFD process, the data collected from the customers are required to be effectively examined

and analyzed to organize and deduce pertinent information (Sharma, Bamnote and Shrivastava, 2003). The data is collected through market research techniques like personal interview, focus groups, surveys, conferencing etc. As a result, a high volume of unsorted and disorganized data is generated, which represent the customer statements. This massive data is required to be transformed into a limited number of structured data to represent the actual 'Customer Needs'. However the process consumes a lot of time and thus in order to save time and effort a suitable technique has to be resorted for the purpose. An Affinity process is one of the techniques used to limit and organize information by splitting data into natural groupings. Although the Affinity process happens to be more of creative than the intellectual one, still software automation of the process can prove to be a useful solution for the QFD practitioners. However development of a software application involves meticulous examination of the process, evaluation of the needed features and then looks for possible implementation with some tools (Sharma, Mehta and Sharma, 2009). The paper thus makes an attempt to present the software framework. In the light of above, the paper is organized as follows. The paper begins with the topic introduction. Section 2 outlines the QFD process. Section 3 discusses Affinity Process. Section 4 discusses the need and benefits and attempts to develop the proposed software framework. It discusses the data structure and considers the evolvment through Visual Basic (VB) and Ms-Access for the proposed framework.

## **2. QUALITY FUNCTION DEPLOYMENT (QFD)**

QFD is based on the widely used Four Phase Model, which focuses on the quality deployment part of Akao's comprehensive QFD framework. The four phases represent product planning, component planning, process planning and production planning (Cohen, 1995). The central tool of the QFD is the matrix chart called House of Quality (HOQ). HOQ is the matrix, which analyses customer requirements in detail and translates them into the developer's language (Griffin, 1992). The QFD comprises several different parts or rooms, which are sequentially filled in order to achieve an actionable translation from requirements into characteristics (Hauser and Clausing, 1988). The first room of the HOQ concerns the Voice of the Customer - a structured list of requirements concerning the product and its attributes as the customer describes them also known as needs, or wants together with a measure of the importance customers attach to each requirement (Govers, 1996). After their identification, the customers' needs are compiled and organized by the QFD team in a hierarchical structure of primary, secondary and tertiary requirements using quality tools like

Affinity Diagram (Sullivan, 1988). The establishment of the voice of the customer is the most critical step in a QFD project.

### **3. AFFINITY PROCESS**

#### **3.1 Limiting the Number of Requirements**

On account of the massive data generated in QFD, the complexity of dealing and managing the data poses a great challenge. Moreover, as the number of input grows, the task becomes more and more onerous. Although the number of requirements varies for each project, still 20-30 items can be considered a good compromise between grade of detail and complexity (Terninko, 1995). The fewer the voices, the easier it will be to work. However as the matrix grows, the consensus decision process will suffer and the overall value of the QFD process and its output will be diminished. For that reason it is obviously important to limit the number of customer requirements. The Affinity Diagram Process is used to further decreasing the number without diminishing the impact of the customers' voice. The best approach to the issue of limiting the matrix size is to use the affinity diagram concept (Day, 1993).

#### **3.2 The Affinity Diagram Process:**

The Affinity Diagram is an analysis tool that allows a team to systematically generate a large volume of ideas or inputs about a problem or issue and organize these into logical groupings. Also called the KJ method, after its developer Kawakita Jiro, an affinity diagram helps to synthesize large amounts of data by finding relationships between ideas. The information is then gradually structured from the bottom up into meaningful groups (Siliconfareast, 2009). Affinity diagrams are an efficient method of organizing large amounts of unstructured verbal data. They enforce a high degree of organization and screen-hidden relationships between the items and item groups. The affinity diagram is one of the Seven New Quality Control (QC) Tools being proposed by Mizuno Shigero (Mizuno, 1988). The process of the KJ-method is, when compared to other management tools, creative other than intellectual. Usually, an affinity diagram is used to refine a brainstorm into something that makes sense and can be dealt with more easily. In Seven New QC Tools, Ishikawa recommends using the affinity diagram when facts or thoughts are uncertain and need to be organized, when preexisting ideas or paradigms need to be overcome, when ideas need to be clarified, and when unity within a team needs to be created (Skymark, 2009). The main objective of the application of the Affinity Process in QFD is to group the customer requirements into a structure of primary, secondary, and tertiary requirements. Affinity Diagram Process is applied for the

organization of customer needs. The affinity diagram process includes several steps and the flowchart for the same is given below (Refer Fig.1). In conventional Affinity Process, as a first step all the customer voices or statements are written on the cards. Cards containing similar items are grouped together on the basis of their affinity. After a few cards are in a group, the group can be labeled. 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. Finally the Affinity Diagram is drawn when all groups are checked and labeled. This generated Affinity Diagram resembles a tree structure of customer requirements on a primary, secondary and tertiary level.

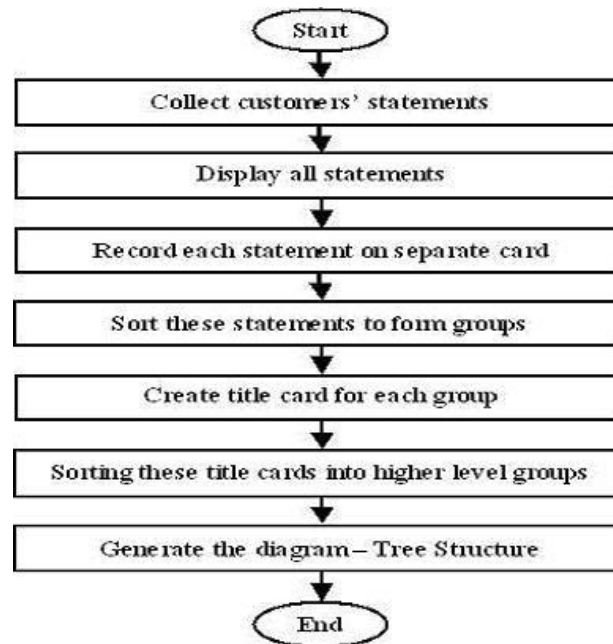


Fig. 1: Flowchart of Affinity Process

### 3.3 An Affinity Example - A Food Supermarket Chain Store:

Affinity Process is a very widely used and powerful technique that deals with massive and disorganized data to produce limited meaningful and organized information. A detailed description of the process and the steps involved are already discussed and presented. An example of a food supermarket chain store adopted from (Syque, 2009) is presented below to further gain insight into the process. The personnel department of a food supermarket chain store identified a

high resignation rate of good checkout staff. There was information available from exit interviews about their reasons for leaving, but this was disorganized and there was no clear area that they felt they could address. They decided to use the KJ method and an Affinity Diagram to try to better understand why these people were leaving (Refer Fig. 2).

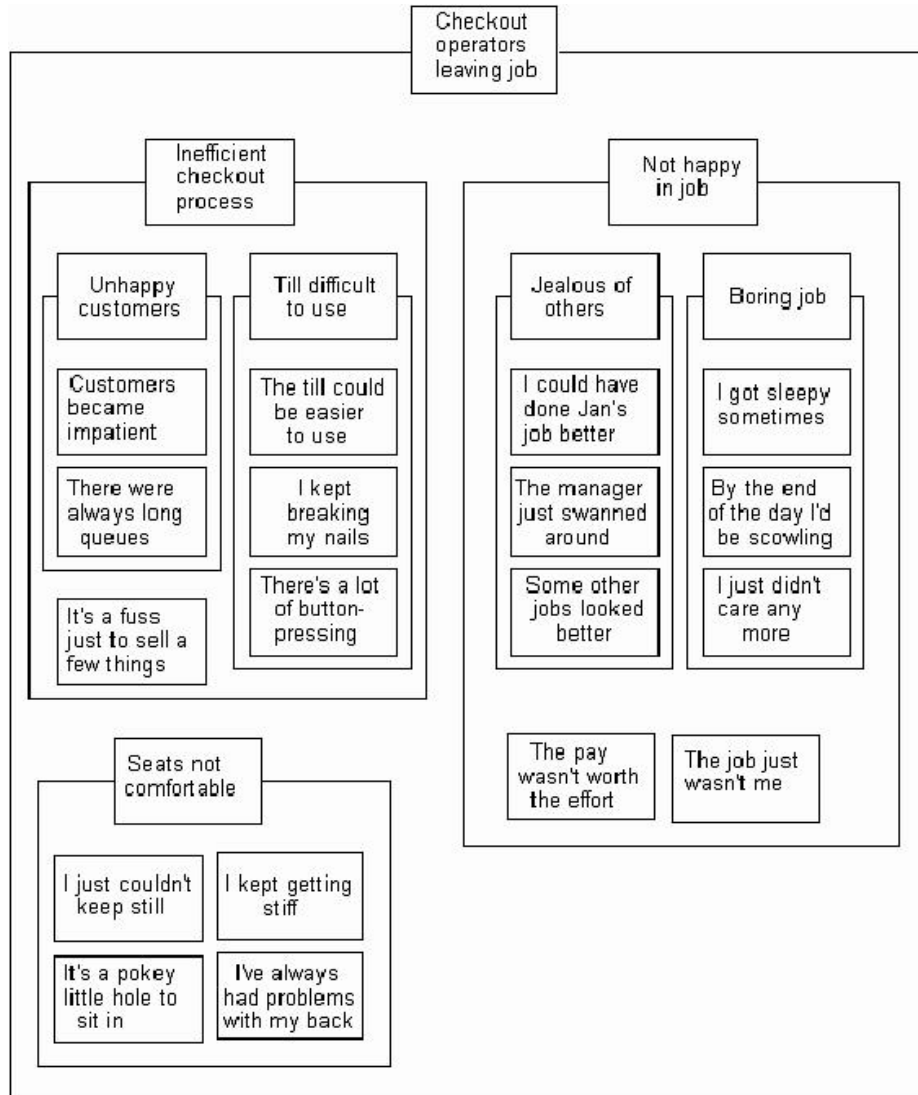


Fig. 2: Supermarket Checkout Example (Source: Syque.com)

As a result, the checkout process was investigated further and eventually completely redesigned. This included a redesigned booth and hourly breaks for operators. Consequently, there were significantly fewer leavers (and as a bonus, customer satisfaction increased).

As seen, the data 'Inefficient checkout process', 'Seats not comfortable' and 'Not happy in job' represent groups while the topmost group titled 'Checkout operators leaving job' can be termed as Superheader. Further the groups also contain subgroups. Thus the process depicts a tree structure represented through blocks. The figure above (Refer Fig. 2) shows the affinity process applied to supermarket chain store. However the inputs to this example are the data in blocks with all groups and subgroups removed. Representing the data in aforementioned fashion could be suitably accomplished using a software system. The proposed framework is presented to serve the purpose. Thus the resultant outcome of the affinity process is a hierarchical tree structure where a 'root' node – 'Checkout operators leaving job' depicts the primary level. The child of the 'root' node represents the secondary level which if further subdivided represents the tertiary level. When done, the elements at the tertiary level are the actual customer needs.

#### 4. A SOFTWARE FRAMEWORK

##### 4.1 Need and Benefits:

In QFD, a large amount of data is collected from the customers, which is to be analyzed and converted to produce actual customer needs. This abundance of the data cannot be dealt manually as the task is tedious and time-consuming and thus a suitable technique is sought for the purpose. Affinity is one process that can be applied in QFD to serve the purpose. The software realization of the process will offer several benefits and greatly assists QFD practitioners in dealing with this massive data. Although the Affinity Process is more of a creative process than the intellectual, yet the process can be software automated to facilitate and increase efficiency of the following aspects:

**Creation of cards:** The Affinity Process makes use of cards that contain customer needs. If the process is software automated, the cards can be printed out, e.g. by using a database system. Moreover this task would no longer be necessary if the process is software automated.

**Conduction of Process:** The completion of an Affinity Process yields a tree structure of customer requirements on a primary, secondary and tertiary level. Of late the programming languages encompass a rich set of controls and are stuffed with many built-in features that assist in software automation. Thus the

process can better be performed through software automation as successfully as using the cards.

**Retrieve Data:** The needs written on cards usually have one or more customer verbatim sources. The voices of the needs cannot be retrieved easily and thus their real meaning i.e. the underlying customer needs, stated by the customer himself, is sometimes lost. Performing the affinity process through the software ensures that the sources of a requirement can be retrieved at any time.

**Save and Recall Data.** The affinity process comprises of several steps that is usually conducted from start to the end. However, there are situations where the process needs to be completed in sections it is necessary to resume the previous work rather than starting it all over again. For this, the data can be saved to the database and then recalled when needed. It is conveniently accomplished if a software tool is used.

**Modify Data:** QFD is a lengthy and complex process that uses a set of matrices. During the process it may require certain set of entered information to be changed. To change the data manually may be onerous; however the same can be effectively carried out through a software tool.

**View Data:** The way affinity diagram is grouped on large table, it is difficult to screen or present the complete diagram. As a result, the complete diagram is viewed in portions. However the issue can be better address through the use of software tool as programming languages offer certain controls that can be programmed. For example VB offers a 'Treeview' control that can be programmed to present any number of data in tree structure.

#### **4.2 Data Structure for Affinity Database:**

To accomplish the software realization of the Affinity Process, an enlargement of the data structure for the tables to be used is required. The table 'tblCust\_Statements' has to be built first to contain the collected information of the customers. The needed data structure should include 'Cust\_St\_Id' to store unique identification number of customers' statements and 'Cust\_St' to store the customer statements. Further, the second table 'tblCust\_Needs' is to be utilized to save the resulted outcome of the process – i.e. tree representation of the actual customer needs. The resultant data is a tree structure and thus has to be saved maintaining the structure. If the structure is maintained the data can be efficiently retrieved from the table and shown in the required form. This table contains 'Cust\_Need\_Id' to store unique identification number of customers' needs, 'Cust\_Need' to store the customer needs, 'Rank' which denotes the level, 'Parent\_Cust\_Need\_Id' to show its parent (the item it is submitted to) and

additionally 'Location' for its position in the list. The location field is not only necessary to save the requirement structure but also to provide a saved storage of inconsistent lists.

The required data structure for the Affinity Database is summarized in tables below (Refer Table 1 and Table2).

<b>Table Name:</b> tblCust_Statements	
<b>Cust_St_Id</b>	<b>Cust_St</b>
AutoNumber-Long Integer	Text

Table1: Customer Statements

<b>Table Name:</b> tblCust_Needs				
<b>Cust_Need_Id</b>	<b>Cust_Need</b>	<b>Rank</b>	<b>Parent_Cust_Need_Id</b>	<b>Location</b>
AutoNumber-Long Integer	Text	Integer	Long Integer	Integer

Table 2: Customer Needs

### 4.3 Software Realization

In order to perform software realization of the process, the process has to be deeply examined, functions and features are evaluated and the possible automation is to be sought. To this end the process has been described and steps involved are discussed. As seen the possible user interface should comprise of two sections as shown on the screen (Refer Fig. 3).

The left section contains the presorted pile of customer statements fetched from the first table 'tblCust\_Statements' that contains the collected customer data to be grouped and structured. The right section is the section where the data from the left section is moved, groups are created, modified, renamed or deleted to generate the structured output i.e. tree structure. The process is completed when all the data from the left section is moved to the right section. When the module is started, a product has to be selected from the dropdown menu (here Teaching Aids) and all customer data from the database table for the chosen product are fetched and shown in left section as 'Total Customer Needs' unless they were previously grouped on the tree. The similar sets of data have to be identified and placed under some group. Initially, when there are no groups in the right section then the group creation takes place at root level, which in our case is 'Teaching Aids' (Refer Fig. 3). This can be done by clicking 'Add' button that on activation

pops up a window to input the group. Like groups, the subgroups can be created using the button 'Add' but it requires a selection of parent group first. For e.g. a subgroup 'Visual Aids' can be created by selecting 'Teaching Aids' first and then clicking 'Add' button which on activation pops up a window to input the subgroup. Finally, once the groups and subgroups are created, the data from the

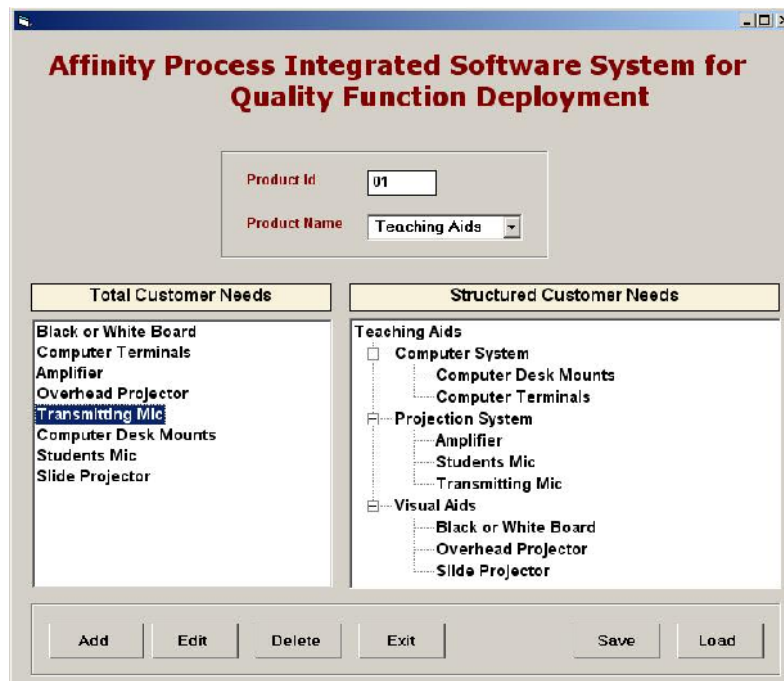


Fig. 3. Proposed Software Interface

left section can be moved one by one and placed under its respective group. For this, the item from the left section is dragged and dropped on the group in the right section. The groups can be created, renamed and deleted using the buttons 'Add', 'Edit' and 'Delete' respectively. Further, the groups and subgroups can be renamed; deleted or new ones can be created at any point of time. The process continues until all items are moved from the left section to the right section. When done, the data can be saved to the database by clicking 'Save' button. However there are times, when the process cannot be completed in go and thus the uncompleted tree structure is saved to the database. In order to complete the structure, this structured saved in database is to be called and displayed for the process to be resumed. The button 'Load' is used to serve this purpose. Finally, the button 'Exit' can be employed to close the application. Moreover for large data, an appropriate way is to be adopted to accommodate all the data for the

display with the tree structure maintained. Thus an appropriate user interface has to be designed. Design is a guide to the function or purpose of each component, while implementation is to create component that has its functions. High quality design should have characteristics that lead to quality products, ease of understanding, ease of implementation, ease of testing, ease of modification, and correct translation from the requirements specification (Ahmad and Wei, 2006). VB comes with a rich set of controls that assists in developing interactive user-interface (Li, 2006). VB is a complete form of package for building user interface (Hassan, Abolarin and Jimoh, 2006). Database interactions for storing and retrieving the data can be achieved using a database tool. Ms-Access is a powerful database system that offers several functionalities related to database tasks and has been widely used as a backend tool for developing several applications. Thus, the software system can be suitably designed and built using VB and Ms-Access. An example of 'Teaching Aids' is considered and its input parameters and generated output are shown in the left and right sections respectively (Refer Fig. 3).

## 5. CONCLUSIONS

The research carried out through this paper attempts to present an Affinity Process integrated software framework for QFD to produce the sorted and structured information. The proposed software framework integrates the Affinity Process in QFD to generate the desired outcome. An efficient and modern software system would greatly assist designers and engineers involved in QFD in dealing with the massive data collected from the customers to produce meaningful information. To this end, in this paper, an attempt is made to build the affinity based software framework for QFD. In this regard, the need and benefits are discussed, the process is analyzed from the software realization perspective, the functioning of the system is exhibited and the possible computer support is discussed. The required data structure is evolved and possible implementation with Visual Basic and Ms-Access is discussed. It is realized that the software system can be suitably designed and built using VB and Ms-Access. Moreover, the proposed framework also makes it possible to extend it using additional modules. We put forth here that the framework discussed has opened an efficient way for developing a powerful and viable software system to cater to the needs of QFD practitioners. It is anticipated that the proposed framework when completely developed will greatly save time and effort and prove to be a valuable asset for QFD practitioners.

**REFERENCES:**

1. Ahmad, M.A. and Wei, L.K. (2006) Development of essol-v2 for engineering survey calculations, *International Journal of Engineering and Technology*, Vol. 3, No.1, 2006, pp. 54-60.
2. Cohen, L. (1995) *Quality Function Deployment: How to make QFD work for you*, J W Wesner (Ed) Reading, MA: Addison-Wesley Publishing Company.
3. Day, R.G. (1993) *Using Quality Function Deployment in the Product Development Process*, *Journal of Applied Manufacturing Systems*, 1993.
4. Govers, C. P.M. (1996) What and how about Quality Function Deployment (QFD), *International Journal of Production Economics*, Vols. 46-47, pp. 575-585.
5. Griffin, A. Evaluating QFD's use in U.S. firms as a process for developing products, *Journal of Product Innovation Management*, Michigan, USA, 1992.
6. Hassan, A.B., Abolarin, M. S. and Jimoh, O. H. (2006) The Application of Visual Basic Computer Programming Language to Simulate Numerical Iterations, *Leonardo Journal of Sciences*, Issue 9, pp. 125-136.
7. Hauser, J. R. and Clausing, D. P. (1988) The House of Quality, *Harvard Business Review*, Vol.66, (5/6) pp. 63-73.
8. King, B. (1989) 'Better designs in half the time - Implementing QFD: Quality Function Deployment in America', 3rd Ed Methuen MA: GOAL/QPC.
9. Li, D. (2006) Design and implementation for wound measurement application, *Primary Intention*; 14(2): 56-58, 60-63, 66.
10. Mizuno, S. (1988) *Company-Wide Total Quality Control*, Asian. Productivity Press, Cambridge.
11. Sharma, A.K., Mehta, I.C. and Sharma, J.R. (2009) Development of Fuzzy Integrated Quality Function Deployment Software – A Conceptual Analysis, *I-Manager's Journal on Software Engineering*, Vol. 3, No. 3, pp. 16-24, 2009.
12. Sharma, A.K., Mehta, I.C. and Sharma, J.R., Analyzing Programming Tools for the Development of Quality Function Deployment Software, *International Journal of Information and Decision Sciences*, Inderscience Publishers, Accepted.
13. Sharma, J.R., Bamnote, A.J. and Shrivastava, R.L. (2003), 'Understanding quality function deployment - A TQM tool to quantify customers needs', *National conference on world class manufacturing*, Coimbatore.
14. Siliconfareast.com, Affinity Diagram, [WWW document] <http://www.siliconfareast.com/affinity.htm> (accessed 03rd April 2009).
15. Skymark.com, Affinity Diagram, [WWW document] [http://www.skymark.com/resources/tools/affinity\\_diagram.asp](http://www.skymark.com/resources/tools/affinity_diagram.asp) (accessed 04th April 2009).
16. Sullivan, L.P. (1988) Policy management through QFD, *Quality Progress*, Vol. 21, No. 6, 1988.
17. Syque.com, Quality Tools: Affinity Diagram Examples, [WWW document] [http://syque.com/quality\\_tools/toolbook/Affinity/example.htm](http://syque.com/quality_tools/toolbook/Affinity/example.htm) (accessed 25th Nov 2009).
18. Terninko, J. (1995) *Step by Step: Customer Driven Product Design*, Nottingham, New Hampshire, USA.