



Development Of An Affinity Process Integrated Software System For Quality Function Deployment

Ashish K. Sharma^{*1}, Sangita A. Sharma², Pankaj S. Agrawal³

^{*1}Asst. Prof., Manoharbai Patel Institute of Engineering and Technology (MIET), Gondia, Maharashtra, India, ash5000@rediffmail.com

²Asst. Prof., Manoharbai Patel Institute of Engineering and Technology (MIET), Gondia, Maharashtra, India, sangitaasharma1975@gmail.com

³Asst. Prof., SS Institute of Engineering and Technology, Durg, Chhattisgarh, India, p_agrawal123@yahoo.com

Abstract

Quality Function Deployment (QFD) is a customer oriented product development process that develops products based on an assessment of customer needs. In QFD, the data is collected from the customers through several techniques like personal interview, focus groups, surveys, conferencing etc. This generates a massive, unsorted and unstructured data, which is required to be transformed into a limited number of structured information to represent the actual 'Customer Needs'. An Affinity Process is used to refine a brainstorm into meaningful information. In QFD, an Affinity Process can be integrated to produce the desired outcome. However, due to process being tedious and time-consuming, in order to save time and effort, suitable software system has to be employed. Moreover, there are situations where the Affinity Process needs to be completed in sections and thus 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. The paper thus makes an attempt to present an Affinity Process integrated software system for QFD. The software system is developed using Visual Basic that integrates a MS-Access database. The database contains the data that represent collected customer needs. The database also allows for storing the generated results, which can be duly utilized in the later stages of the QFD process. The software system has been designed as a framework, which makes it possible to extend it using additional modules.

Index Terms: Affinity Process, Customer Needs, Database, MS-Access, QFD, Software, Visual Basic.

1. INTRODUCTION

Quality Function Deployment (QFD) comprises of sets of matrices, and is rather a lengthy and complex process, which involves voluminous data and complex calculations. As a result, dealing with the data manually is quite cumbersome and time consuming (Sharma, Mehta & Sharma, 2008). QFD is a customer oriented product development process that strives for higher quality and better customer satisfaction. Being a customer oriented process; customer satisfaction is of utmost importance in QFD. In QFD, 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. Customer attributes must be analyzed carefully before making design decisions. Techniques for identifying customer wants and needs include surveys, clinics, focus groups, individual interviews, listening (eavesdropping) at dealerships and trade shows, consumer complaints, etc. (Day 1993, Dahlheimer 1995). As a result, a high volume of unsorted and disorganized data is generated, which represent the customer statements. Since this data is highly unstructured, in order to bring value to these data, the data has to be distilled and suitably converted to represent the actual customer needs. 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). To this end, the similar data is

figured out and effectively grouped to identify a small number of representative needs. However, since the collected data happens to be voluminous, grouping and identification is tedious and time-consuming. These calls for a suitable technique that can better address this issue. Affinity Process is a widely used and powerful method used to organize and gain insight into a set of qualitative information, such as voiced customer needs. The Affinity Diagram is one of the Seven New Quality Control (QC) Tools being proposed by Mizuno Shigeru (Mizuno, 1998). Also called the KJ method, after its developer Kawakita Jiro, the process of the KJ-method is, when compared to other management tools, creative other than intellectual. Usually, an initial list of customer attributes is developed internally through the brainstorming in most of the QFD projects. Affinity Diagrams are then used to consolidate and organize the individual attributes. Application of Affinity Process on collected customer data in QFD yields a tree structure of customer needs on a primary, secondary and tertiary level. When the tree is completed, the voices on the third level or any other level with a limited number of requirements are extracted. These requirements are the final requirements that will be used in the House of Quality (HoQ) of QFD. However due to the complexity and time-consuming nature of the method, need of a suitable software is highly recognized. The paper thus attempts to develop a software system, which can synthesize and organize the

massive data generated in QFD to produce meaningful information. The software system is developed using Visual Basic and MS-Access. The database contains the data that represents collected customer needs. The database also allows for storing the generated results, which can be duly utilized in the later stages of the QFD process. An example is presented to demonstrate the application of the software system. The outcome of the work is an efficient and viable software tool that integrates Affinity Process to produce a clear segmentation of customer needs. These needs symbolize the 'WHAT' section of HOQ.

In the light of above, the paper is organized as follows. The paper begins with the topic introduction. Section 2 presents a literature review on application of an Affinity Process in QFD. Section 3 briefly outlines the QFD process. Section 4 describes the Affinity Process. Section 5 presents the research methodology to be used. Section 6 describes the proposed software system. Finally the last section presents the conclusions of the developed software system.

2. QFD AND AFFINITY PROCESS: A LITERATURE REVIEW

Affinity Process is a vital tool that deals with massive data to produce limited number of organized and meaningful information. Affinity Process employs Affinity Diagrams, which 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. In QFD also, Affinity Process has been utilized for better and productive results. Affinity Diagrams are applied to structure the lifestyle words in QFD (Mazur, 2005). A method used for transforming the customer requirements, sometimes called the voice of customer (VoC), to product specifications was developed by (Shoji et al., 1993), in which semantics methods, such as the Affinity Diagram and MPM (multipickup method), are applied as the basis for discovering underlying facts from affective language. (Cruz & Tamayo, 2003) applied QFD for tackling internal customers' needs as a base for building a lean manufacturing system. Affinity Diagrams are employed for grouping customer needs into different categories in building a lean manufacturing system. (Antony, 2004) carried out a pilot survey in the UK service organizations to understand the status of Six Sigma and 40 variables i.e. statements were grouped under 13 critical success factors (CSFs) using the Affinity Diagram tool. (Valtasaari, 2000) employed the Affinity Diagrams in QFD to design customer needs for product development at Nokia Mobile Phones (NMP). (Fung & Popplewell, 1995) proposed preprocessing the VoC prior to its being entered as customer attributes into the house of quality. In preprocessing, they adopted an Affinity Diagram (KJ method) to categorize the customer requirements.

3. QUALITY FUNCTION DEPLOYMENT (QFD)

QFD is a customer-oriented approach, supporting design teams in developing new products based on an assessment of customer needs. Basically, in the QFD, customer needs are translated into design attributes. To begin the design process in the QFD, the design team needs to listen to the voice of the customer. 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). QFD is an innovative approach bringing quality as demanded by the customers – upstream into the product development process. QFD is a systematic process for helping a business to focus on its priorities, investments and 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 methodology provides a way to translate conceptual requirements into items that are workable, measurable, and capable of design enhancement. The result is a better design, shorter product development cycles, better product quality, and lower costs (Lin et al., 2004). The central tool of the QFD is the matrix chart called House of Quality (HOQ). This tool is a powerful way of generating specific, prioritized, and measurable technical requirements from often-ambiguous customer needs. The HOQ matrix is the most recognized form of QFD. It is utilized by a multidisciplinary team to translate a set of customer requirements into an appropriate number of prioritized engineering targets to be met by a new product design. 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 & 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). There are several possible sources of information about customer requirements: market research data, sales data, customer complaints, retailers, focus groups, opinion surveys, in-depth interviews, etc. The customers' requirements are compiled and organized by the QFD team in a hierarchical structure of primary, secondary and tertiary requirements using quality tools like Affinity Diagram. The establishment of the voice of the customer is the most critical step in a QFD project.

4. AFFINITY PROCESS

4.1 Restricting Number of Needs

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).

4.2 Affinity Diagram Process

The Affinity Process is formalized in an Affinity Diagram and is useful when sifting through large volumes of data and encouraging new patterns of thinking. An Affinity Diagram is a tool that gathers large amounts of language data ideas, opinions and issues and organizes them into groupings based on their natural relationships. The Affinity Process is often used to group ideas generated by brainstorming. Affinity Diagrams are the tools used to consolidate and organize the individual attributes. Customer attributes must be analyzed carefully before making design decisions. The voice of the customer should be organized into a hierarchy. 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.com, 2004). 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. 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 is then converted into a tree structure of customer

needs on a primary, secondary and tertiary level. A Tree Diagram can be applied for the purpose.

The Affinity Process includes several steps and the flowchart for the same is given below (Refer Fig.1):

4.3 Flowchart Of Affinity Process

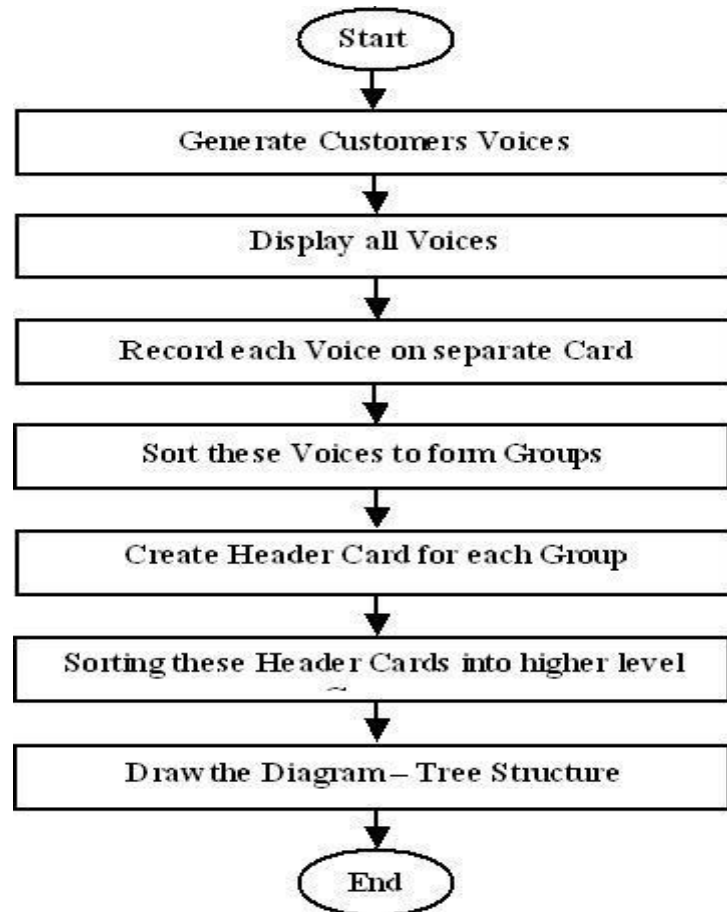


Figure 1: Process Flow of Affinity

5. METHODOLOGY

The goal is to develop a customized Affinity Process based software system to cater to the needs of QFD. Development of an application enforces meticulous examination of the processes involved, gathering of data and then transcribing these data into the technical slabs for effective utilization (Sharma, Mehta and Sharma, 2009). Thus, the methodology encompasses suitably analyzing the process from the development perspective, eliciting out the requirements, evaluating features to be implemented and then look for possible implementation of the same using the selected systems. Since Affinity Process is a multi step process that makes use of creation of elements and moving elements. Besides the software realization of the process, it also accounts for accommodating and screening all data when grouped on larger tables. Thus, an appropriate interface is to be designed and developed. Further, since the results or data

generated shall be duly utilized for future use and in the latter stages of the QFD process, a suitable database system is to be used. VB includes a lot more functionality than most other software packages. VB is a very powerful system that is stuffed with many ActiveX controls, features and functionalities, which can be suitably utilized to create applications on the fly. VB is considered for the development task. Analyzing the requirements and expected features of the proposed model necessitates the use of front end and back end tool. VB was derived from BASIC and enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using Data Access Objects (DAO), Remote Data Access objects (RDO), or ActiveX Data Access Objects (ADO), creation of ActiveX controls and objects. The use of VB has been stated in a number of articles (Beaty, 1999; Dunham, Sherba & Rutherford, 1996; Pelechano & Darias, 2000; Saeki, Uchida & Ito, 1998). MS-Access is powerful database system that offer several functionalities related to database tasks and has been widely used as backend tool for developing several applications. In this regard, VB and MS-Access are considered as a front-end and back end tool for the development of the proposed model. The user interface and the coding part are achieved through the VB while MS-Access is utilized for the back end purpose.

6. SYSTEM DESCRIPTION

Creating a software system using a software engineering process contains three main tasks or phases: the functions and features of the expected software have to be defined, the software has to be implemented and it has to be deployed in an operating environment. The functions and features of the expected software are called requirements (Kivinen, 2006).

This study is designed to see how the proposed software might increase accuracy and reliability. The parameters for software construction included that it is able to:

- Primarily aim at the following primary characteristics: functionality, reliability, usability, efficiency and maintainability
- Provide strong interaction with the database – most importantly for fetching and storing data
- Establish strategies for construction (coding standards and naming conventions).
- Offer great ease of operability
- Customized user interface to meet the requirements
- Provide graphical output
- Provide support for needed update and delete operations
- Minimize error handling.
- Provide security

6.1 User-Interface

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 & 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 et al., 2006). The software interface is designed and built using VB (Refer Fig. 2). This system has been designed as a framework, which makes it possible to extend it using additional modules. The interface makes use of several controls, which serve different purposes. Besides the basic set of controls, the interface also employs ActiveX controls such as 'Listview' and 'Treeview'.

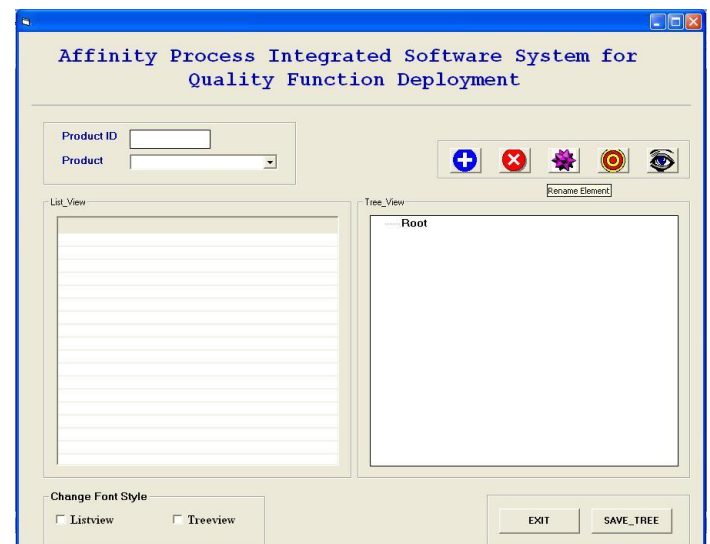


Figure 2: Software Interface

The interface comprises of two sections left and right section that houses the ActiveX controls – 'Listview' and 'Treeview' respectively. A 'Dropdown' is provided to make a selection for the product. The interface employs a frame, which imitates a toolbar and contains a set of graphical buttons to facilitate the organization of elements. The buttons are positioned in sequence from left to right to carry out the following functions - Create, Delete and Rename groups. The last two buttons on the toolbar on right provides customized viewing as 'Normal View' and 'Full Screen View'. In 'Normal View' both the windows are visible while in 'Full Screen View' only 'Treeview' in maximized state is shown to have a better view of structured data. The system also supports viewing contents of 'Listview' and 'Treeview' using different 'Font Names' and 'Font Sizes'. This functionality can be availed through the checkboxes located in 'Change Font Style' frame. To change the 'Font' parameters, the corresponding checkbox is activated which pops up the font dialogue window and the changes are done. Further a 'Save_Tree' button is provided to save the generated structure to the database. A 'Tooltip' text is provided for each object on the interface when the mouse pointer is moved over the object. This helps in understanding the functions associated to each object. Finally, 'Exit' button is also provided to close the

application. VB provides powerful features such as graphical user interface, events handling access to Win 32 API, object- oriented features, error handling, structured programming and much more (Hassan et al., 2006). The software interface developed with VB will offer great compatibility. In addition, with the help of VB, the user-friendly interface can be further improved.

6.2 Code Module

Programming is matching design with implementation. The first step in writing software is formulating the problem and then utilizing the suitable code for implementation. Before the process of software design, the modules of the proposed model have to be identified so that the design would not be aimless. The role of modules defining is to supplement some design guideline during the process of design (Ahmad and Wei, 2006). To this end, the proposed software has been broken into several procedures and implemented in code module. The implementation of the developed user interface has been achieved using two code modules. The code module encloses a software realization of an Affinity Process method. The code module contains all the programming code as a collection of procedures and sub procedures while the basic module contains API's declaration. The codes associated to the click of buttons, drag and drop operations are all incorporated in the code module. In order to increase the width of 'Listview' column so that it can accommodate lengthy text, a set of API's is used in basic module. A snapshot of the code module is shown below (Refer Fig. 3)

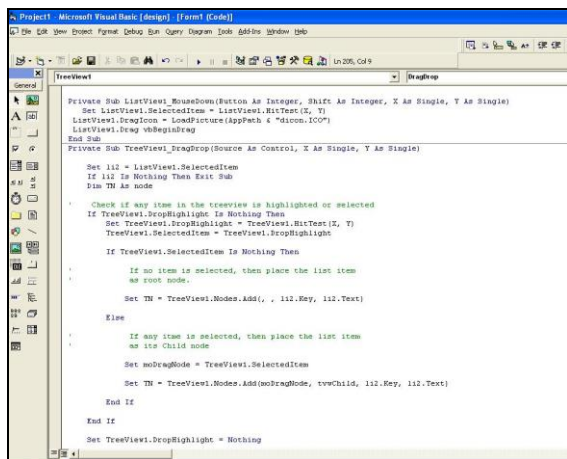


Figure 3: Code Module

6.3 Affinity Database

The back end functionalities are achieved through the MS-Access database tool. The MS-Access database was used to store and interpret the data. MS-Access database is simple to use and contains all the essential features. MS-Access is powerful database tool that offer several functionalities related to database tasks and has been widely used as backend tool for developing several applications. Thus, 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. Data in this table is stored when SaveData button is clicked. The required data structure for the Affinity Database is tabulated in tables below (Refer Table 1 and Table2).

Table Name: tblCust_Statements	
Cust_St_Id	Cust_St
AutoNumber-Long Integer	Text

Table1: Customer Statements

Table Name: tblCust_Needs				
Cust_Need_Id	Cust_Need	Rank	Parent_Cust_Need_Id	Location
AutoNumber-Long Integer	Text	Integer	Long Integer	Integer

Table 2: Customer Needs

6.4 Working

The user interacts with the screen (Refer Fig. 2) above, when the application is set to run mode. The Affinity software system is not only designed to structure needs, it is a rather universal and flexible module that can handle any kind of data, e.g. Design Characteristics, Functions, or Concepts. The Affinity Process comprises of several steps and is usually conducted from start to the end. However, there are situations where the process needs to be completed in sections and thus 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. Thus, the developed system should allow for working in both 'Start-to-End' and 'Resume' modes. As a first step, a

product for which the collected data has to be processed and organized is selected from the dropdown menu and all items i.e. collected customer statements from the database table for the chosen product are fetched and shown in left section that houses 'Listview'. The first step also accounts for checking a pre-existence of created structure for the selected product. If it exists then the structure is loaded from the database and displayed in 'Treeview' section so that the process can be resumed there from. However in the absence of pre-existed structure, only an element with text 'Root' is shown as a default item in the 'Treeview'. The text of this default item can be changed to represent the title or heading of the structure to be generated. The next step is to create some initial group headings on the tree. To create a group heading, select the 'Root' item and then click on the 'Add' button and then enter the text for group heading. This adds the newly created group to the 'Root'. The process is repeated for other groups. Next, the items from the left section can be moved to the added groups on the tree in right section. This is done through a 'Drag and Drop' operation. To move an item from 'Listview' to 'Treeview', select an item, drag this item and drop it on the group in 'Treeview'. This places a "+" sign in front of the group to which the item is added to serve as an indication of existence of child elements for that group. The process continues until all items are moved from the left section to the right section. The text of items and headings can be changed; deleted or new ones can be created. Further, items can be moved from one group to another or arranged within their group in the tree. The group can be expanded or collapsed by clicking on the sign in front. The "+" sign in front of the group depicts the collapsed mode of the group and thus items under the group cannot be visible. In order to browse these items, the "+" sign is clicked which switches it to expanded mode. This mode is represented by "-" sign. Thus a large number of items can be accommodated in the tree and the items are expanded when required. Additionally, it is possible to move entire groups to another position within the tree structure. The system even allows moving elements 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. This is done by 'Drag and Drop' function but with the right mouse button. The added items can be deleted and renamed at any point of time. This is done by selecting the item and clicking the graphical buttons 'Delete' and 'Rename' respectively. The system also facilitates viewing in two modes 'Normal View' and 'Full Screen View'. In 'Normal View', both 'Listview' and 'Treeview' are visible while in 'Full Screen View'; 'Treeview' is expanded and is only visible. This is required when all the items are added to the tree and so the 'Listview' is empty and thus the 'Treeview' can be maximized to have better view of structured items. The system also supports viewing contents of 'Listview' and 'Treeview' using different 'Font Names' and 'Font Sizes'. This functionality can be availed through the checkboxes located in 'Change Font Style' frame. To change the 'Font' parameters, the corresponding checkbox is activated which pops up the font dialogue window and the changes are done. Some of the working of the features is shown in snapshots below (Refer Fig 4, 5 and 6).

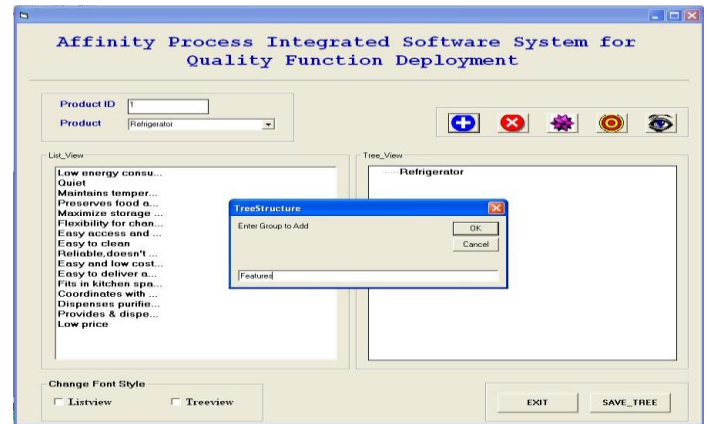


Figure 4: Add Element

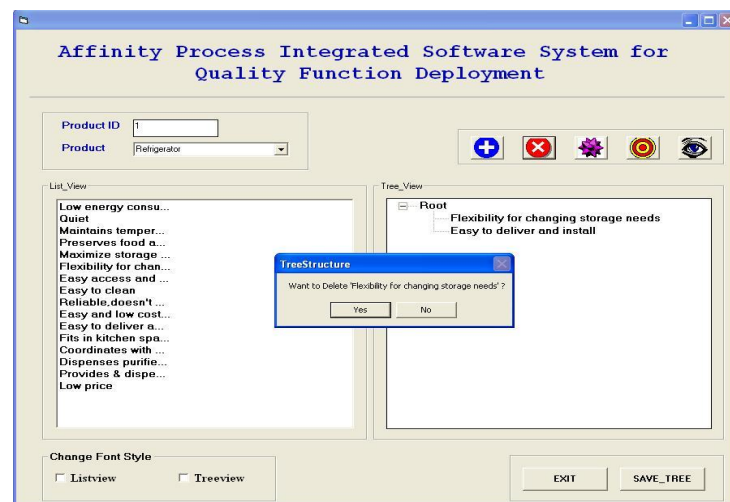


Figure 5: Delete Element

CONCLUSIONS

A suitable Affinity Process integrated software system for QFD will not only reduce the unnecessary onus but also assists in producing results in more effective and accurate manner. To this end, this paper presents an Affinity Process integrated software system to cater to the needs of QFD practitioners. In this paper, the development and implementation of the prototype system are discussed in details. The system is developed using Visual Basic 6.0 that integrates an MS-Access database. The use of Visual Basic with MS-Access database is found to be very effective in producing the system under windows environment. Besides the software realization of Affinity Process, the system incorporates certain features like 'Full Screen View', 'Change Font Style' and 'Save Tree', which add new dimension to the tool. The system also facilitates all the needed operations for creating and organizing items like add, rename, move and delete. Further, a set of application programming interface (API) is employed to accommodate larger text items in 'Listview' as no direct function is available. Moreover, in QFD there are situations where at times the Affinity Process cannot be completed in one go and thus 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. The developed system also facilitates this feature. Moreover, the software system has been designed as a framework, which makes it possible to extend it using additional modules. The system is a very user friendly and efficient, which would offer great ease of operability, comfort and would significantly save time and effort. This software system specifically designed for QFD will greatly assists designers

REFERENCES

- 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.
- Antony, J. (2004) 'Six Sigma in the UK service organizations: results from a pilot survey', *Managerial Auditing Journal*, Vol. 19, No. 8, pp. 1006-1013.
- Beaty, J.C. (1999) 'The Power Checker: a Visual Basic Program for Ensuring Data Integrity', *Behavior Research Methods, Instruments & Computers*, Vol.31, pp.737-740.
- Cohen, L. (1995) 'Quality Function Deployment: How to make QFD work for you', J W Wesner (Ed) Reading MA: Addison-Wesley Publishing Company.
- Cruz J. S. and Tamayo F. (2003) 'QFD application for tackling Internal Customers Needs as a base for building a Lean Manufacturing System', *Lean Manufacturing, The 9th International Symposium on QFD (ISQFD'03) and The 15th Symposium on QFD*.
- Dahlheimer, Per (1995), *The Customer Input Process*, Sydney, 1995
- Day, R. G. (1993), *Quality Function Deployment - Linking a Company with its Customers*, Milwaukee, Wisconsin, USA, 1993
- Dunham, D.W., Rutherford, P. and Sherba, M. (1996), ' INFOBITS: A Visual Basic Program for Calculating Information Theoretic Statistics from Behavior Sequences', *Behavior Research Methods, Instruments & Computers*, Vol.28, and pp. 622-626.
- Fung, RT. K. & Popplewell .K. (1995) The analysis of customer requirements for effective rationalization of product attributes in manufacturing, in: *Proceedings of 3rd International Conference on Manufacturing Technology*, Hong Kong.
- Griffin, A. (1992) 'Evaluating QFD's use in U.S. firms as a process for developing products', *Journal of Product Innovation Management*, Michigan, USA.
- Govers, C. P. M. (1996) 'What and how about Quality Function Deployment (QFD)', *International Journal of Production Economics*, Vols. 46-47, pp. 575-585.
- 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, July-December 2006, pp. 125-136.
- Hauser, J. R. and Clausing, D. P. (1988) 'The House of Quality', *Harvard Business Review*, Vol.66, (5/6) pp. 63-73.
- King, B. (1989) 'Better designs in half the time - Implementing QFD: Quality Function Deployment in America', 3rd Ed Methuen MA: GOAL/QPC.
- Kivinen, T (2006) *Applying QFD to improve the requirements and project management in small-scale project*, University of Tampere, Department of Computing Sciences, Computer sciences.
- Li, D. (2006) 'Design and implementation for wound measurement application', *Primary Intention*; 14(2): 56-58, 60-63, 66.
- Lin M.C., Tsai, C.Y., Cheng C.C. and Chang C.A (2004) 'Using Fuzzy QFD for Design of Low-end Digital Camera', *International Journal of Applied Science and Engineering*, 2004. 2, 3: 222-233.
- Mazur G.H. (2005) 'Lifestyle QFD: Incorporating emotional appeal in product development', *The 17 symposium on QFD*, Portland.
- Mizuno, S. (1998) *Company-Wide Total Quality Control*, Asian. Productivity Press, Cambridge.
- Pelechano, V. and Darias, E. (2000) 'Computerized Personality Questionnaires In: Microsoft Visual Basic 6.0 for Windows', *Psicothema*, Vol.12, pp. 418-422.
- Saeki, D., Uchida, Y. and Ito, M. (1998) 'A Visual Basic-based Control System for Behavioral Experiments', *Japanese Journal of Behavior Analysis*, Vol.13, pp. 66-72.
- Sharma, A. K., Mehta, I.C. and Sharma, J.R. (2008), 'Analyzing Fuzzy Quality Function Deployment (QFD) – A Software Automation Perspective'. *National Conference on Recent Innovative Techniques in Engineering and Management*, Oct 4-5, 2008. HCET, Jabalpur.
- Sharma, A.K., Mehta, I.C. and Sharma, J.R. (2009) *Development of Fuzzy Integrated Quality Function Deployment Software – A Conceptual Analysis*, *Journal of Software Engineering*, Accepted.

Shoji, S., Graham, A. & Walden, D. (1993) A New American TQM (Portland, OR, Productivity Press).

Siliconfareast (2004)
<http://www.siliconfareast.com/affinity.htm>

Skymark (2009),
http://www.skymark.com/resources/tools/affinity_diagram.asp

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

Valtasaari, M. (2000) 'Design for Customer Needs: Utilization of Quality Function Deployment in Product Development', Masters Thesis: Lappeenranta University of Technology.