APPLYING QFD TO IMPROVE REQUIREMENT AND PROJECT MANAGEMENT IN RUBRICS TECHNOLOGIES PVT. LTD.

[1]Mr. Saikiran Ravi Chindam, [2]Dr. Lavendra S. Bothra
Alamuri Ratnamala Institute of Engineering and Technology (ARMIET)[1,2]

ABSTRACT

Quality Function Deployment (QFD) is a customer-driven planning methodology used to translate customer needs into design requirements, ensuring product and service quality. This paper explores the application of QFD in enhancing requirement and project management processes at Rubrics Technologies Pvt. Ltd. By aligning project goals with customer expectations, QFD helps streamline workflows, improve communication, and reduce project risks. The study highlights the integration of QFD tools like the House of Quality to identify key customer needs, prioritize technical solutions, and track performance metrics. The results demonstrate significant improvements in project timelines, resource allocation, and overall client satisfaction. The findings suggest that adopting QFD can lead to more efficient project management and better alignment between project deliverables and customer expectations.

Keywords: Quality Function Deployment (QFD), Project Management, Requirement Engineering, House of Quality

INTRODUCTION:

Creating a software system using a software engineering process contains three main tasks or phases: the functions and features of the expected the software has to be defined, the software has to be implemented and it has to be deployed in an operating environment.

Most software systems are developed as project since a software system is usually unique and the software should be produced in a certain time limit and resources. The sizes of projects vary a lot. There are also a lot of different methods and different process models to manage the projects but the common feature is that phases mentioned above are always included in to a software system development project even though they usually are divided and specified into more detailed tasks.

The functions and features of the expected software are called requirements. Requirement analysis is the phase of software development where feasibilities studies are made, competitors and existing systems are examined and the new system is specified.

In spite of new and effective software engineering techniques, according to various researches a lot of software development projects tend to fail.

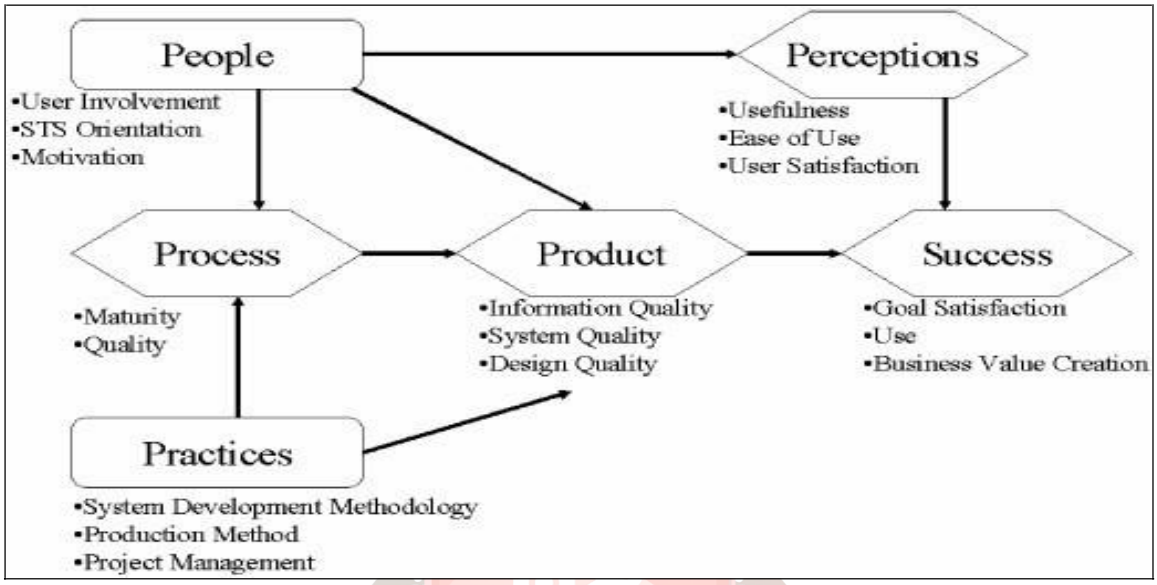
The three main reasons for the failure in projects were:

1. Lack of User Input
2. Incomplete Requirements & Specifications
3. Changing Requirements & Specifications

Main Concepts and the Structure of the Study:

The most important concepts used in this study and their relationships are illustrated in Figure 1.

Figure 1: Information Systems Quality Model



The most important concept is the quality of software. There are a lot of factors which define the quality of software: reliability, usability, etc. and also multiple perceptions of quality. There are also other of factors and forces which collectively affect the quality of software. These factors and their multidimensional effect on each other are illustrated in Figure 1.

Software Development Processes:

A process framework establishes the foundation for a complete software process by identifying a small number of framework activities that are applicable to all software projects, regardless of their size or complexity. The main activities are applicable to the vast majority of software projects and they are:

- Communication with stakeholders which includes requirement acquisition
- Planning
- Modeling
- Construction
- Deployment

Software engineering incorporates a development strategy that encompasses the process, methods, and tools. This strategy is often referred to as a software process model or a software engineering paradigm. A software development process can also be defined as a set of activities needed to transform the user requirements into a software system.

The process models divided into two types: prescriptive and agile software development. Prescriptive software process models prescribe a set of process elements: framework activities, software engineering actions, tasks, work products and quality assurance and change control mechanisms for each project. Each process model also prescribes a workflow, that is, the manner in which the process elements are interrelated to one another.

All prescriptive process models accommodate the earlier mentioned main activities, but each applies a different emphasis to these activities and defines a workflow that invokes each framework activity as well as software engineering actions and tasks in a different manner. There are a lot of different kinds of development processes and some of them are just combination of features from the others. The prescriptive process models classified into five types: waterfall models, incremental process models, evolutionary process models, specialized process models and the rational unified process (RUP).

Traditional process models tend to be inflexible to changes which helps them maintain a predictable schedule, but it does nothing to ensure that the final results meets the customers real, changing needs. Agile software development aims to flexibility and adjustability. It emphasizes individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation and responding to change over following a plan. Commonly used agile process model types are Extreme Programming (XP), Scrum, Adaptive Software Development, Feature Driven Development and DSDM but there are several others too.

For example most agile process types are using features of incremental and evolutionary processes.

RESEARCH PROBLEM:

The goal is to develop a good understanding of the QFD method. Moreover, this study discovering QFD tools that can assist engineers to develop high quality products

The QFD method has its origins in manufacturing industry, where it has been applied successfully. The success is probably due to some characteristics of QFD that make it effective. Understanding these characteristics could help in making QFD successful also in software engineering.

Scope of the Study:

This study centers on the quality function deployment method. The scope of the study is using QFD in software engineering. QFD was originally developed for the manufacturing industry to produce high quality products. There have been many attempts to use QFD in the development of software products; however, software and hardware have quite different characteristics, and this must be taken into account before applying a method used to manufacture product such as QFD into developing software products.

OBJECTIVES:

To find out how much extra work does it entail in small scale project.

1. To deploy QFD in a small-scale project.
2. To help the project manager to make decisions during the project.
3. To improve project quality.

LIMITATIONS:

This report might be limited with some constraints. Details of many aspects of the services of Rubrics Technologies Pvt. Ltd. have been skipped in this report due to various constraints, including time and space. Moreover, one of the main barriers in writing this report was the confidentiality of data. Besides, for the preparation of this report I found that there are some limitation exists though I have given my greatest effort. Such as:

1. The employees are very busy with their job which lead me little time to consult with.

2. Category wise annual turnover, strategic group map, amount originated from the international trade is missing in the report for their restriction.
3. In many cases, up to date information was not published.
4. Non-availability of reference books is another limitation.
5. Due to time limitation many of the aspects could not be discussed in the present report. Learning all the functions within just 60 days is really tough.

REVIEW OF LITERATURE

Concept of Quality:

Quality is a concept that lacks a clear and concise definition and is thus difficult to accurately measure, improve or even compare across different industries, products and services. The quality of a service refers to the extent to which the service fulfills the requirements and expectations of the customer.

An understanding of the basic concepts on quality and its management is essential for the professional management of Quality of Service (QoS). According to ISO 8402 standard the concept of quality can be defined as "totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs". An entity is an item that can be individually described and considered. An entity may be an activity or a process, a physical product or an organization, actually anything that has a quality attribute.

Quality is a phenomenon; it is an emergent property of people's different attitudes and beliefs, which often change over the development life-cycle of a project.

The concept of quality can be observed different standpoints. There are six different viewpoints. A customer oriented viewpoint emphasizes customer satisfaction, which arises from fitness for use. A manufacturing oriented viewpoint means similarity and consistency in the manufacturing process. A product oriented viewpoint emphasizes product performance and value oriented viewpoint concentrates on getting best possible quality for the lowest possible price. Additionally an environment oriented viewpoint emphasizes the environmental impact of the product.

A basic element of quality is that it is not free: it always requires efforts typically reviews, testing, inspections etc. which cost but on the other hand it always adds some value to the customer. Experiences in manufacturing relating to the cost and return of quality improvements suggest that there are diminishing returns to quality expenditures. Therefore the key management problem is how to make profitable decisions on quality expenditures. Typically quality characteristics may be required or not, or may be required to a greater or lesser degree, and trade-offs may be made among them.

Software quality:

Software of good quality can be defined as software without any errors and deficiencies. Yet it is very difficult to prove that software doesn't contain any errors. Thus software of good quality is software without anyknown errors and deficiencies.

The business value of a software product results from its quality. Quality is increasingly seen as a critical attribute of software, since its absence results in financial loss as well as dissatisfied users, and may even endanger lives.

Since the importance of setting software quality requirements and assessing quality is better recognized a shift from creating technology centered solutions is made to satisfying stakeholders. Software acquisition, development,

maintenance and operations organizations confronted with such a shift are, in general, not adequately equipped to deal with it. Until recently, they did not have the quality models or measurement instruments to allow or facilitate the engineering of quality throughout the entire software product life cycle. The objective of software product quality engineering is to achieve the required quality of the product through the whole production process: definition of quality requirements and their implementation, measurement of appropriate quality attributes, and evaluation of the resulting quality. The objective is, in fact, software product quality.

Software quality can be divided into three main categories or topics: software quality fundamentals, software quality management processes and practical considerations as in Figure below.

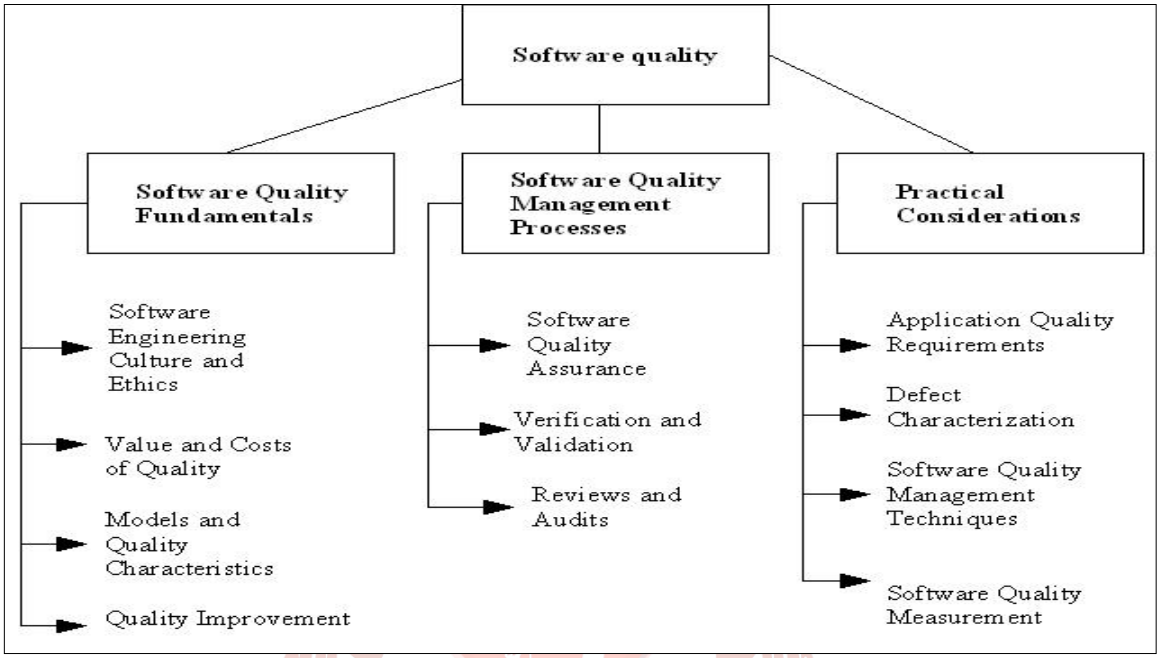


Figure: Categories of Software Quality

CONCLUSION:

In the beginning, five objectives were stated for this Study. The study contributes to creating understanding about the quality function deployment method by presenting the concepts and procedures of QFD. The study describes the QFD method and how it is applied in the industry. The description covers the basic elements of QFD and the phases of the application process. Literature sources are used to give examples of QFD variants and approaches in different application areas including software engineering. QFD is also compared to some other decision-support methods.

The Study contributes to summarizing the preconditions for successful application of QFD. The study discusses the meaning of success and gives an overall analysis of QFD, while creating an understanding of what QFD actually promises and what is necessary in order to keep those promises. The observations and conclusions are derived from the literature.

The study contributes to the third objective of gathering experiences of QFD in real-life projects. The study presents direct observations from QFD applications.

Finally, the study contributes by giving suggestions to a company about things that might be worth considering in utilizing and applying QFD in projects. The guidelines are synthesized from information found in literature

sources, experiences from the analyzed cases, and general observations from real-life software development projects.

REFERENCES:

1. Managing Small Projects, Simon Buehring. IT toolbox Research White paper, 2006.
2. QFD Application in Software Process Management and Improvement Based on CMM, Xiaoqing (Frank) Liu, Yan Sun and Gautam Kane, 2005.
3. A Software Development Process for Small Projects, E. Trengove and B. Dwolatzky. International Journal of Electronical Engineering Education, 2004.
4. A Quantitative Approach for Setting Technical Targets Based on Impact Analysis in Software Quality Function Deployment (SQFD), Frank Liu, Kunio Noguchi, Anuj Dhungana, V.V.N.S.N. Srirangam A. and Praveen Inuganti,. 2006.