9. NPD strategies #1

9.1 Design for Six Sigma

Design for Six Sigma (DFSS) is a separate and emerging business-process management methodology related to traditional Six Sigma. While the tools and order used in Six Sigma require a process to be in place and functioning, DFSS has the objective of determining the needs of customers and the business, and driving those needs into the product solution so created. DFSS is relevant to the complex system/product synthesis phase, especially in the context of unprecedented system development. It is process generation in contrast with process improvement.

There are different options for the implementation of DFSS. Unlike Six Sigma, which is commonly driven via DMAIC (Define - Measure - Analyze - Improve - Control) projects, DFSS has spawned a number of stepwise processes, all in the style of the DMAIC procedure. Another option is, however, to integrate the DFSS approach into the Product Development Process.

DMADV, define – measure – analyze – design – verify, is sometimes synonymously referred to as DFSS. The traditional DMAIC Six Sigma process, as it is usually practiced, which is focused on evolutionary and continuous improvement manufacturing or service process development, usually occurs after initial system or product design and development have been largely completed. DMAIC Six Sigma as practiced is usually consumed with solving existing manufacturing or service process problems and removal of the defects and variation associated with defects. On the other hand, DFSS (or DMADV) strives to generate a new process where none existed, or where an existing process is deemed to be inadequate and in need of replacement. DFSS aims to create a process with the end in mind of optimally building the efficiencies of Six Sigma methodology into the process before implementation; traditional Six Sigma seeks for continuous improvement after a process already exists.

9.1.1 DFSS as an approach to design

DFSS seeks to avoid manufacturing/service process problems by using advanced Voice of the Customer techniques and proper systems engineering techniques to avoid process problems at the outset (e.g., fire prevention). When combined, these

methods obtain the proper needs of the customer, and derive engineering system parameter requirements that increase product and service effectiveness in the eyes of the customer and all other people. This yields products and services that provide great customer satisfaction and increased market share. These techniques also include tools and processes to predict, model and simulate the product delivery system (the processes/tools, personnel and organization, training, facilities, and logistics to produce the product/service) as well as the analysis of the developing system life cycle itself with proper investigation results and gains to ensure absolute customer satisfaction with the proposed system design solution. In this way, DFSS is closely related to systems engineering, operations research (solving the knapsack problem), systems architecture, workflow balancing, and concurrent engineering and even more. DFSS is largely a design activity requiring specialized tools including: quality function deployment (QFD), axiomatic design, TRIZ, Design for X, design of experiments (DOE), Taguchi methods, tolerance design, robustification and Response Surface Methodology for a single or multiple response optimization. While these tools are sometimes used in the classic DMAIC Six Sigma process, they are uniquely used by DFSS to analyze new and unprecedented systems/products.

9.1.2 Distinctions from DMAIC

Proponents of DMAIC and Lean techniques might claim that DFSS falls under the general rubric of Six Sigma or Lean Six Sigma. It is often seen that the tools used for DFSS techniques vary widely from those used for DMAIC Six Sigma. In particular, DMAIC practitioners often use new or existing mechanical drawings and manufacturing process instructions as the originating information to perform their analysis, while DFSS practitioners often use system simulations and parametric system design/analysis tools to predict both cost and performance of candidate system architectures. While it can be claimed that two processes are similar, in practice the working medium differs enough so that DFSS requires different tool sets in order to perform its system design tasks. DMAIC Six Sigma may still be used during depth-first plunges into the system architecture analysis and for "back end" Six Sigma processes; DFSS provides system design processes used in front-end complex system designs.

Traditional six sigma methodology, DMAIC, has become a standard process optimization tool for the chemical process industries. However, it has become clear that the promise of six sigma, specifically, 3.4 defects per million opportunities (DPMO), is simply unachievable after the fact. Consequently,there has been a growing movement to implement six sigma design usually called design for six sigma DFSS. This methodology begins with defining customer needs and leads to the development of robust processes to deliver those needs.

9.1.3 Similarities with other methods

Arguments about what makes DFSS different from Six Sigma demonstrate the similarities between DFSS and other established engineering practices such as probabilistic design and design for quality. In general Six Sigma with its DMAIC roadmap focuses on improvement of an existing process or processes. DFSS focuses on the creation of new value with inputs from customers, suppliers and business needs. While traditional Six Sigma may also use those inputs, the focus is again on improvement and not design of some new product or system. It also shows the engineering background of DFSS. However, like other methods developed in engineering, there is no theoretical reason why DFSS can't be used in areas outside of engineering.

9.1.4 Software engineering applications

Historically, although the first successful Design for Six Sigma projects in 1989 and 1991 predate establishment of the DMAIC process improvement process, Design for Six Sigma (DFSS) is accepted in part because Six Sigma organisations found that they could not optimise products past three or four Sigma without fundamentally redesigning the product, and because improving a process or product after launch is considered less efficient and effective than designing in quality. ‘Six Sigma’ levels of performance have to be ‘built-in’.

DFSS for software is essentially a non-superficial modification of "classical DFSS" since the character and nature of software is different from other fields of engineering. The methodology describes the detailed process for successfully applying DFSS methods and tools throughout the software product design, covering the overall Software Development life cycle: requirements, architecture, design, implementation, integration, optimization, verification and validation

(RADIOV). The methodology explains how to build predictive statistical models for software reliability and robustness and shows how simulation and analysis techniques can be combined with structural design and architecture methods to effectively produce software and information systems at Six Sigma levels.

DFSS in software acts as a glue to blend the classical modelling techniques of software engineering such as object-oriented design or Evolutionary Rapid Development with statistical, predictive models and simulation techniques. The methodology provides Software Engineers with practical tools for measuring and predicting the quality attributes of the software product and also enables them to include software in system reliability models.

9.2 Flexible product development

Flexible product development is the ability to make changes in the product being developed or in how it is developed, even relatively late in development, without being too disruptive. Consequently, the later one can make changes, the more flexible the process is, the less disruptive the change is, the greater the flexibility.

Flexibility is important because the development of a new product naturally involves change from what came before it. Change can be expected in what the customer wants and how the customer might use the product, in how competitors might respond, and in the new technologies being applied in the product or in its manufacturing process. The more innovative a new product is, the more likely it is that the development team will have to make changes during development.

Flexible development counteracts the tendencies of many contemporary management approaches to plan a project completely at its outset and discourage change thereafter. These include Six Sigma, which aims to drive variation out of a process; lean, which acts to drive out waste; and traditional project management and phased development systems (including the popular Phase–gate model), which encourage upfront planning and following the plan. Although these methodologies have strengths, their side effect is encouraging rigidity in a process that needs flexibility to be effective, especially for truly innovative products.

For more mature product categories, flexibility techniques are not only overly expensive but often unwise. Consequently, flexibility techniques must be used with discretion, for instance, only in the portions of a product likely to undergo change.

When applied to the development of software products, these methods are commonly known as agile software development. However, agile software methods generally rely on special characteristics of the software medium, especially object technologies, which are not available to non-software products. Consequently, flexible product development draws from some of the roots of agile software development but tends to use other tools and approaches that apply beyond the software medium.

Flexible development uses several techniques to keep the cost of change low and to make decisions at the last responsible moment. These techniques include modular architectures to encapsulate change, experimentation and iteration to sample results and check them out with the customer frequently, set-based design to build and maintain options, and emergent processes that develop during a project in response to its needs.

9.3 Quality function deployment

Quality function deployment (QFD) is a "method to transform user demands into design quality, to deploy the functions forming quality, and to deploy methods for achieving the design quality into subsystems and component parts, and ultimately to specific elements of the manufacturing process.", as described by Dr. Yoji Akao, who originally developed QFD in Japan in 1966, when the author combined his work in quality assurance and quality control points with function deployment used in value engineering.

QFD is designed to help planners focus on characteristics of a new or existing product or service from the viewpoints of market segments, company, or technology-development needs. The technique yields charts and matrices.

QFD helps transform customer needs (the voice of the customer [VOC]) into engineering characteristics (and appropriate test methods) for a product or service, prioritizing each product or service characteristic while simultaneously setting development targets for product or service.

9.3.1 Areas of application

QFD is applied in a wide variety of services, consumer products, military needs, and emerging technology products. The technique is also included in the new ISO 9000:2000 standard which focuses on customer satisfaction.

While many books and articles on "how to do QFD" are available, there is a relative paucity of example matrices available. QFD matrices become highly proprietary due to the high density of product or service information found therein.

9.3.2Techniques and tools based on QFD

9.3.2.1 House of Quality

House of Quality appeared in 1972 in the design of an oil tanker by Mitsubishi Heavy Industries. Akao has reiterated numerous times that a House of Quality is not QFD, it is just an example of one tool.

A Flash tutorial exists showing the build process of the traditional QFD "House of Quality" (HOQ). (Although this example may violate QFD principles, the basic sequence of HOQ building are illustrative.) There are also free QFD templates available that walk users through the process of creating a House of Quality.

Other tools extend the analysis beyond quality to cost, technology, reliability, function, parts, technology, manufacturing, and service deployments.

In addition, the same technique can extend the method into the constituent product subsystems, configuration items, assemblies, and parts. From these detail level components, fabrication and assembly process QFD charts can be developed to support statistical process control techniques.

9.3.2.2 Pugh concept selection

Pugh Concept Selection can be used in coordination with QFD to select a promising product or service configuration from among listed alternatives.

9.3.2.3 Modular Function Deployment

Modular Function Deployment uses QFD to establish customer requirements and to identify important design requirements with a special emphasis on modularity. There are three main differences to QFD as applied in Modular Function Deployment compared to House of Quality:

The benchmarking data is mostly gone.

The checkboxes and crosses have been replaced with circles.

The triangular “roof” is missing.

There are also other minor differences between the application of QFD in Modular Function Deployment as compared to House of Quality, for example the term "Customer Attribute" is replaced by "Customer Value", and the term "Engineering Characteristics" is replaced by "Product Properties". But the terms have similar meanings in the two applications.