INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

A COMPARISION OF SYSTEMATIC DESIGN AND DESIGN FOR SIX SIGMA

Larry Stauffer[1] and Tushar Pawar[2]
[1]Professor, University of Idaho
[2]Graduate Assistant, University of Idaho

ABSTRACT
In this paper we provide a broad look into Systematic Design and Design for Six Sigma, two methodologies for designing technical systems and products. The Systematic Design approach is well structured and documented and has a long history of success in the marketplace. The Design for Six Sigma approach is broader, has many different versions, and is still evolving. Both approaches accomplish the same thing: customer driven products. But when one examines the tools and methods used in the major phases of both approaches, there are differences. In this paper we have highlighted these differences. We explain that both Systematic Design and Design for Six Sigma could benefit from an exchange of tools and methods.

Keywords: Systematic Design, Six Sigma, Design for Six Sigma

1 INTRODUCTION

For the past several decades Systematic Design (SD) has been developed and accepted as the best practice for product design. This approach provides a widely documented process of turning customer requirements into a working product. At a high level, SD begins with a problem or need. The methodology presents a clear path through planning, conceptual, embodiment, and detailed design. In all of the phases SD is a fairly prescriptive process, providing excellent guidance to the designer.
Six Sigma is a structured approach for eliciting and applying data to improve or design a product, service, or process. It is composed of two programs. The first program, called Six Sigma Improvement (SSI), is focused on improving a product, service, or process. The approach follows the DMAIC (Define, Measure, Analyze, Improve, and Control) model. The second program is focused on designing new rather than improving upon an existing product, service, or process. It is known as Design for Six Sigma (DFSS). Unlike SSI, DFSS has many approaches. SD, SSI, and DFSS will be discussed in more detail later in this section.

1.1 Objective

In this paper we provide an overview of both Systematic Design and Design for Six Sigma. Then the differences and commonalities of the two approaches are detailed. It is our contention that both can benefit from the influence of the other.

1.2 Overview of Systematic Design

Systematic Design (SD) is a methodology for developing technical systems or products. It is as much of a field of information and guidance as it is a step by step methodology. It is well documented, most notably in the books <u>Engineering Design</u> [1] and <u>Design of Technical Systems [2]</u>, and VDI Design Handbook 2221 [3] as well as numerous other publications. To adequately describe the process would require referencing these and thousands of published papers and books, which is beyond the scope of this paper. For the sake of simplicity (and to make this section practical) we reference only one work unless otherwise noted: <u>Engineering Design</u> written by Gerhard Pahl and Wolfgang Beitz as translated by Wallace, Blessing, and Bauert [1]. Even though a third edition has just been printed we reference the second edition as it is in wide use and is the same regarding the topic of this paper. This book is perhaps the most widely referenced work on SD and includes the most widely understood topics and terminology. Taking this "short cut" will help us focus more on Six Sigma which is not as well formed and less widely known to the design research community.

SD became formalize after World War II, primarily in the 1960's, 70's, and 80's, though some of the concepts go back to the days of Leonardo da Vinci. One may view SD as a European work as the vast majority of contributors are from there; northern Europe in particular. Yet there have been other contributions, primarily over the past 20 years, from other parts of the world that have added to the body of knowledge. SD is in wide use, in whole or in part, especially for the design of mechanical or electro-mechanical products.

SD is typically presented as having four main phases: product planning, conceptual design, embodiment (or layout) design, and detail design. The separation between one phase and the next is not always firm and there is often backtracking as new information becomes clearer. But in general, these phases provide a roadmap for following the SD methodology. Product planning begins with a product idea that holds promise given the current market and economy along with the needs of the company or designer. There are also times when a product idea comes to the company as a specific request from a customer. In either case, the task at hand must be clarified through intense market/customer analysis and known constraints, resulting in a product proposal and requirements list.

During conceptual design, the requirements list gets transformed into a conceptual, principle solution. This phase is perhaps the most challenging, unique, and important phase of SD. A key aspect of this phase is to develop a function structure. A function structure demonstrates how the incoming energy, materials, and information are transformed by the product. In this way we can develop a solution without making firm commitments of a detailed, physical nature. Only after consideration of many possible alternatives does the design group make a commitment to a single principle solution.

During the embodiment phase the principle solution is developed into a more definitive layout of the technical product. During this time technical and economic criteria are used to transform the concept into a very real and practical solution. Specific issues such as performance, safety, ergonomics, manufacture, and other life-cycle issues are addressed. SD categorizes these issues as rules, principles, and guidelines. Through this process, the very nature of the product is specified, as well as all of its components and capabilities. As the design team considers all of these issues new concepts and changes may need to be considered. Significant backtracking may occur even to the point of discussing the problem with customers again. The end result is a definitive layout of the technical system that will meet customer expectations.

During the final phase of detail design, the product becomes a reality. Each component is uniquely characterized, dimensions and tolerances are set, materials are specified, and performance is assured. System performance as a function of component performance is analyzed. Business issues are also addressed. The design team typically estimates the expected cost of the design and production drawings are finalized. The output of this phase becomes the specification for production.

In summary, SD follows a four phase approach: product planning, conceptual design, embodiment, and detail design. While this description is good for helping to understand SD the task is rarely this simple. There is much iteration bouncing back and forth between phases. During embodiment for example, a design change may be identified while assessing ergonomics which calls for additional conceptualizing. But overall, these four phases provide a good description of the SD process.

1.3 Overview of Six Sigma

Six Sigma is a methodology to improve the capability of business processes such as a manufacturing process, a service process, or an internal process such as billing or order entry [4]. A simpler definition of is Six Sigma is reducing variation to increase process performance. The increase in performance leads to defect reduction and improvement in profits. But more than a methodology, Six Sigma is an approach towards delivering products and services with high performance as measured by critical to satisfaction (CTS) metrics. Six Sigma has been credited with saving billions of dollars for companies since the early 1990's. Six Sigma was begun by Motorola Corporation in the mid 1980's. It became well known only after General Electric Company made it a central focus of its business strategy in 1995 [5]. The name "Six Sigma" derives from statistical terminology; sigma means standard deviation. For a normal distribution, the probability of falling within a +/- six sigma range around the mean is 0.9999966. It assumes one can perform with only 3.4 defects per million opportunities for a defect [6].

Six Sigma uses several proven design and quality tools and methods, most of which have been in use for decades. A Six Sigma program has two main parts. First is Six Sigma Improvement (SSI); the other is called Design for Six Sigma (DFSS). When an organization says they are practicing Six Sigma they are referring to SSI. The primary method in SSI is a simple performance improvement model, often called a roadmap, known as Define-Measure-Analyze-Improve-Control (DMAIC). One can think of this roadmap as a five phase process for conducting Six Sigma. During the Design phase the team creates a charter or summary of the project, makes a detailed problem statement, maps the primary process steps, and identifies independent and dependent variables. For the Measure phase they collect data to characterize the process in detail. During Analyze they use the characterization to best understand process performance and how changes in the independent variables affect the dependent variables. This information is applied during the Improve phase to determine target values and a robust solution. The Control phase focuses on how to maintain the improvements. By following this roadmap processes can experience dramatic improvement as systems operate more consistently. This strategy however ***does not*** involve any changing or redesigning of the fundamental structure of the underlying process. It involves finding solutions to eliminate the root causes of performance variation problems, while leaving the basic process intact.

1.4 Overview of Design for Six Sigma

Design for Six Sigma (DFSS) goes upstream and requires changing or redesigning the fundamental structure of the underlying process [4]. DFSS provides for the original design of a product, service, or process. It is not a strategy to improve the current situation but to provide a fundamental change in the structure of the product, service, or process. DFSS can be applied to the design of electromechanical products (the main thrust of SD), but also to the design of software systems, transactional services, operational processes, and so forth. It can be said that DFSS is gaining acceptance because of SSI. But it is also resonating with industry. In recent years DFSS has been gaining traction in America and is seeing an ever increasing use in Europe, Asia, and the rest of the world.

The purpose of DFSS is to “design it right the first time” so that constant improvements are unnecessary. As discussed in the previous section, DFSS also has the assumption of 3.4 defects per million opportunities. That is, if a product is designed following DFSS a parameter’s value will have so little variation that the upper and lower specification limits will be plus or minus six standard deviations. This level of performance is only a goal, of course. No methodology for such a complex process can guarantee such high levels of performance. There are many versions of DFSS (discussed below). All of them derive from, or work in tandem with, SSI. In the study of these various approaches, one is presented with a roadmap and various tools and best practices that can be applied during each phase. There is little if any theoretical basis for DFSS with the exception of Yang and El-Haik [6]. They provide a theoretical basis of DFSS based upon the areas of quality engineering [7], TRIZ [8], axiomatic design [9], and probability and statistical modeling. They present their theory from the perspective of *vulnerabilities*. They think of six sigma variation as the level at which design vulnerabilities are not effective or at least minimal. There are two major design vulnerabilities that may affect the quality of a designed entity (product, service, or process) [6]

- Conceptual vulnerabilities, established because of violation of design axioms and principles.
- Operational vulnerabilities, due to lack of robustness in the use environment.

The common SSI approach addresses the operational vulnerabilities by trying to eliminate or reduce their impact. DFSS addresses both conceptual and operational vulnerabilities. Operational vulnerabilities take variability reduction and mean adjustment of critical-to-quality (CTQ) requirements, as an objective. Hence, tolerance research is at the heart of operational vulnerabilities as it deals with the assignment of tolerances in the design parameters and process variables, assignment of control and manufacturing processes, the metrological issues, as well as the geometric and cost models. Conceptual vulnerabilities on the other hand are usually overlooked because of the lack of a systematic and disciplined approach to find ideal solutions, ignorance of the designer, schedule deadlines pressure, and budget limitations. Partly, this is attributable to the fact that traditional quality methods which can be best described as ***after-the-fact*** practices use lagging information to developmental activities such as bench tests and field data. These practices drive design towards endless cycles of “***design-test-fix-retest***”. They contend that most DFSS approaches simply take the standard DMAIC roadmap and incorporate a more extensive voice-of-the-customer (VOC) component. Corrective actions to improve the conceptual vulnerabilities via operational vulnerability improvement means are only marginally effective. Implementing Six Sigma tools and practices early in the conceptual phase is therefore imperative. Attention thus begins to shift from improving the performance during the later phases of the design life cycle to the upfront phases where product development takes place at a higher level of abstraction. This shift is also motivated by the fact that the design decisions made during early stages of the design life cycle have the greatest impact on total cost and quality of the system. Often 80 percent of the total cost is committed in the product development phase [10].

2 DESIGN FOR SIX SIGMA APPROACHES

Like SSI, DFSS has the use of a roadmap at its core. However, there is not one accepted roadmap, but many. With the growing popularity of DFSS practices, different DFSS methodologies have evolved with time and some are still evolving. The literature is full of their acronyms: DMADV, DMADOV, ICOV, CDOV, IDOV, DMEDI, DCCDI, and DCOV to name some. Deploying companies of the Six Sigma philosophy devise their in-house views of DFSS. Many times a company implements DFSS to suit their business, industry and culture; other times they implement the version of DFSS used by the consulting company assisting in the deployment.

The most popular roadmap among those mentioned above is the DMADV- "Define-Measure-Analyze-Design-Verify" [4].

Define the project goals and customer (internal and external) requirements.

Measure and determine customer needs and specifications; benchmark competitors and industry.

Analyze the process options to meet the customer needs.

Design the process to meet the customer needs.

Verify the design performance and ability to meet customer needs.

A slight modification on the DMADV methodology is DMADOV: Define Measure, Analyze, Design, Optimize and Verify [4]. This roadmap collects and emphasizes optimization activities.

Another roadmap is ICOV: Identify-Characterize-Optimize-Verify [6].

Identify the customer and critical-to-satisfaction (CTS) requirements.

Characterize the design by translating the customer CTSs into functional requirements. Then generate solution alternatives and evaluate them to determine the "best" solution.

Optimize uses advanced statistical tools and modeling to predict and optimize the design and performance.

Validate makes sure that the resulting product will meet the customer CTSs.

DCCDI, another DFSS methodology [11].

Define the project goals.

Customer analysis is completed.

Concept ideas are developed, reviewed and selected.

Design is performed to meet the customer and business specifications.

Implementation is completed to develop and commercialize the product/service.

IDOV [12], is similar to ICOV, and is put forward as:

Identify the customer and specifications (CTQs).

Design translates the customer CTQs into functional requirements and into solution alternatives. A selection process whittles down the list of solutions to the "best" solution.

Optimize uses advanced statistical tools and modeling to predict and optimize the design and performance.

Validate makes sure that the design you've developed will meet the customer CTQs.

We have listed only five of the several DFSS roadmaps. Though the phases may be different most of the tools used within these phases are similar. Quality Function Deployment (QFD) [13], Pugh's Concept Selection [14], and Failure Modes and Effects Analysis (FMEA) [15] are just a few of the tools used in DFSS. There are no tools that were actually developed for DFSS. Rather it has simply collected tools from other programs and combined them into the DMADV or similar roadmap. DFSS also has a strong emphasis on statistics, in terms of how data is represented and experiments are designed and analyzed.

How does one decide among the different DFSS roadmaps for designing products? Unlike the DMAIC methodology, the phases and steps of DFSS are not universally recognized or defined. All roadmaps are applicable to a product, process or service design however some are better suited for product design than others. In general the difference between product and service development is the level of detail and complexity of the tools used in the Optimize phase of the DFSS project. ICOV and CDOV are applicable to both product design and service design however the tools that are emphasized in these roadmaps favor product design more than service design because of the emphasis on optimization. The DMADV roadmap, being the most popular, can be undertaken for all process design and service design as well as a product design. However DMADOV, with an extra "O", emphasizes optimization and is better suited for product design than DMADV.

3 COMPARING SYSTEMATIC DESIGN AND DESIGN FOR SIX SIGMA

To begin this discussion we identify a roadmap for SD; PCED: Plan-Conceptualize-Embody-Detail. At first it might seem odd to characterize SD with the PCED roadmap but this characterization helps to compare SD and DFSS in a similar fashion. To represent DFSS we primarily rely on the ICOV roadmap as its phases have similar purposes to PCED. An immediate reaction for comparing both roadmaps is to take a systematic approach; develop evaluation criteria, rank order the two approaches for each criteria, add up the results, and declare a winner. But that is not the purpose of this paper and nor could we find agreement on such a comparison. Rather, we take the perspective of identifying similarities and differences between the two approaches. Furthermore, we will mention only a few of the more important comparisons for each phase of the roadmaps. To make a detailed comparison would require more text than this paper allows. Unless otherwise noted, all the design tools and methods can be found in reference [1] for SD and reference [6] for DFSS.

3.1 Roadmaps Plan and Identify

In both roadmaps, the purpose of Plan from PCED (for SD) and Identify from ICOV (for DFSS) are the same: define the product and create a requirements list. The steps of *Plan* are: analyze the market and company needs, find and select product ideas, formulate a product proposal, clarify the task, and build a requirements list. The steps of *Identify* are: draft a project charter and identify customer and business requirements. When one looks at what these steps accomplish, they are similar. The difference lies in the techniques. SD supports a wide variety of planning tools, matrix methods, and checklists. DFSS supports fewer tools, namely QFD, the Kano Model, benchmarking (in the context of QFD), and the Klein Model. It should be noted that neither approach precludes mixing of tools and techniques. It is our observation that DFSS tools provide more detail regarding customer information and requirements. For example, SD separates requirements into *demands* and *wants* whereas DFSS separates them into *needs*, *wants*, and *delights* and further categorizes them as to customer priority from low to high impact. The benchmarking phase of QFD helps a company understand how they are ranked with the competition. Conversely, SD encourages more completeness, especially in the use of checklists to make sure all areas of concern are addressed. Following a complete checklist from SD while conducting QFD in conjunction with the Kano and Klein models could yield an effective approach. There is precedence for this approach in past research [16].

3.2 Roadmaps Conceptualize and Characterize

The second phases of the roadmaps are Conceptualize (SD) and Characterize (DFSS). Here the two approaches tend to converge but at slightly different end-points. The end-point of the *Conceptualize* phase is a principle solution or concept that adequately describes the very nature of the product. The end-point of the *Characterize* phase adds two more steps that so not appear until the start of SD's *Embody* phase. The steps and techniques employed by both roadmaps are different and are demonstrated in Figure 1 below.

DFSS begins with a study to understand the evolution of the functional requirements (FRs). In this step a few of the FRs that are CTS are identified and studied as to how their value has changed over time and when new technologies have created dramatic shifts in their values. The classic example is the evolution of land speed and how the technology shifts from horse and buggy, to steam engine, to automobile created large shifts in land speed. The TRIZ methodology [8] is used to aid this task. SD could benefit by including this step in some cases as long as a time/benefit estimate is first made.

Conceptualize (SD)	Characterize (DFSS)
	Understand FR evolution
- Abstract to identify essential problems - Establish function structures - Search for working principles - Combine working principles - Select suitable combinations - Firm up into principle solution variants	- Generate and analyze concepts
- Evaluate variants against criteria	- Select best concept
- *Complementary effort during the Embody phase*	- Finalize the physical structure of selected concept - Perform mappings

Figure 1 Comparison of Steps in Conceptualize and Characterize Phases

DFSS has a coupled step, to generate and analyze concepts by the method of Controlled Convergence [14], taking advantage of Axiomatic Design [9] and TRIZ. Controlled Convergence calls for the alternating of concept generation and selection steps until a good solution is converged upon. The generation activity is enhanced by the use of TRIZ and axiom 1, from Axiomatic Design, which calls for functional independence. The selection activity is enhanced by axiom 2 which calls for simplicity. The end result of this step is a morphological matrix of a set of alternative solutions that are ready to be evaluated; which is also a point in the Conceptualize phase of SD (see Figure 1). For product design, we view the related six steps, from the *essential problem* to *principle solution variants,* to be a more logical and helpful towards developing alternative solutions. The effort spent developing the function structure results in an excellent depiction of the relationship between requirements and solutions that is missing from DFSS. But rather than just dismiss DFSS at this point, there could be some advantage in utilizing TRIZ during the search for working principles. There could also be advantage in using axioms 1 and 2 as criteria when examining the suitability of alternative concepts. Borrowing these techniques could enhance SD.

From the morphological matrix, alternative concepts are examined to select the best alternative. SD uses Use-Value-Analysis, also known as Cost-Benefit-Analysis and DFSS uses Pugh’s concept selection technique. The former is more comprehensive and the latter is simpler and faster. Both approaches work though the merits of both can be debated.

DFSS ends the Characterize phase with two key steps that are not included in the Conceptualize phase of SD. These steps may be considered as quality checks on the concept before the design process moves on to the latter phases. The first of two steps is to finalize the physical structure of the selected concept. The purpose is to ensure adherence to the independence and minimal information axioms. If the selected alternative solution violates these axioms there may be limited success during use. The second step of mapping is to make preliminary designs of processes for producing the product. Creative manufacturing methods are needed to increase the likelihood of high quality, robustness, and controllability. DFSS views these last two steps as a major weakness of SD. It claims that these last two steps must be pursued to minimize conceptual and operational vulnerabilities discussed in section 1.4 above. These vulnerabilities lead to products that do not meet six sigma quality levels. We believe this criticism of SD to be ill founded, for the axioms of independence and minimal information are similar in concept to the *rules* of simplicity and clarity during the Embody phase. Perhaps one could claim that DFSS is deficient because it does not include the third rule of *safety*. But since safety concerns are addressed latter, we consider the point to be mute.

3.3 Roadmaps Embody and Optimize

The third phases of the roadmaps are Embody (SD) and Optimize (DFSS). Here the two approaches tend to converge at slightly different end-points. It should be recalled that the rules of *clarity* and *simplicity* have the same aim as the final steps of the Characterize phase in DFSS. As in section 3.2 above, the steps and techniques employed by both roadmaps are different. Many of the differences however, are in terminologies as the activity is the same. (For example, *Principles for Embodiment* *design* and *Design for X issues* both describe the need and techniques to address the various life-cycle issues of ergonomics, manufacture, etc.) The activities for these two approaches are listed in Figure 2 below.

Embody (SD)	Optimize (DFSS)
- 15 steps of Embodiment design	- Uncouple or decouple concept
- Checklist for Embodiment design	- Simplify design
- Design fault (error) identification	- Scorecards & Transfer functions
- Rules of Embodiment design	- Risk assessment
- Principles for Embodiment design	- Transfer function organization
- Guidelines for Embodiment design	- Design for X issues
	- Tolerance design

Figure 2 Comparison of Activities in Embody and Optimize Phases.

The Embody phase of SD refers more to a list of design activities than a phase with a step-by-step process. This phase entails everything that must be done to get the product concept ready to be detailed for production. At the end of this phase there must be a clear understanding of performance, production, and cost. While the Conceptualize phase is methodical with some iteration from beginning to end, the Embody phase calls for constant iteration between synthesis and analysis until optimized, detailed definition of the product is obtained. Many activities are done simultaneously and details are checked and corrected as new information comes into view. Pahl and Beitz [1], our datum reference, lists a 15 step process for this phase but it is meant to be more of a guideline depending on the exact nature of the product being designed. It is not possible to fully describe the Embody phase in the space of this paper; the reader is referred to Chapter 7 of our datum reference. The Embody phase provides a checklist of life-cycle issues (ergonomics, production, maintenance, etc.) that must be addressed. It also provides a set of issues which must be considered as the steps and checklist are addressed. These issues are listed as rules (necessary considerations for all products) as well as principles and guidelines (necessity depends on the nature of the product).
Some of the activities in the Embody and Optimize phases are similar. An activity of the Optimize phase is to *simplify the design* which is one of the three rules of the Embody phase. *Design fault identification* and *risk assessment* both call for the use of Failure Modes and Effects Analysis [15]. *Guidelines for embodiment design* are essentially *design for X issues* and include *tolerance design*.
There are numerous differences in the two approaches. Due to space limitations we can not cover all of them but will highlight one unique difference which is the use of *transfer functions* and *scorecards* in DFSS. A transfer function is a relationship that links factors in the mappings between customers, product, and process. For every CTS requirement, there is a transfer function that links functional requirements to design parameters and another that links design parameters to process variables. This concept is demonstrated in Figure 3.

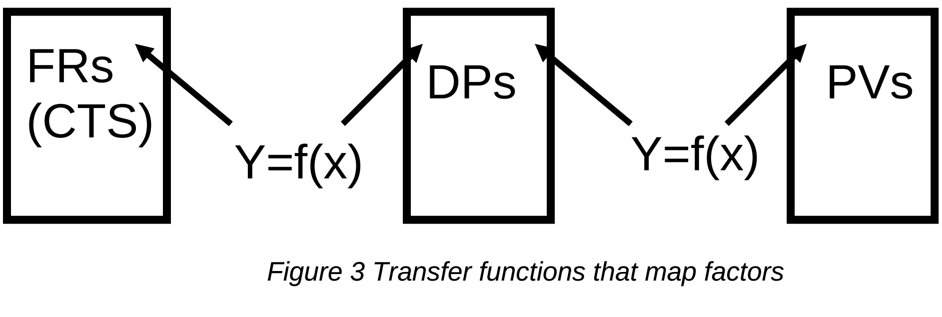

Figure 3 Transfer functions that map factors

Transfer function relationships are preferably mathematical, derived from physical principles but can be obtained empirically from statistically designed experiments. If mathematical formulas can not be produced, the relationship can be modelled. There should be a transfer function for every CTS. For example, the set of transfer functions that relate FRs to DPs can be defined in matrix form as:

$$\{FRs\} = [A]\{DPs\} \qquad (1)$$

The independent variables, Xs, can have their mean values and/or variances adjusted in response to product and process noise factors so as to optimize the dependent variable, Y, in the transfer function. This optimization migrates through the system of transfer functions which leads to higher customer satisfaction. The *scorecard* is a visible document for recording and assessing design progress as transfer functions are used to adjust the factors during the optimization process. It is used to communicate the status of the effort and also identify gaps in information, predict final results, and assess progress.

It should be noted that the call for optimization is not unique to DFSS. SD also calls for optimization throughout the roadmap yet the call does not play as significant of a role. The Optimize phase is primarily focused on addressing a full consideration of concurrent optimizations of CTS requirements as they migrate through the process from customer to product to production. We believe many design efforts using SD could benefit from this focus. One weakness of DFSS however, is the lack of detail and completeness provided by SD for further defining the concept. The various rules, principles, and guidelines provide a wealth of guidance on which DFSS is silent. This is especially true for non-performance issues such as appearance, safety, usability, and so forth. Addressing these types of issues is usually not done in the same quantitative format as issues of functionality. They are better served by design rules, principles, and guidelines.

3.4 Roadmaps Detail and Validate

The final phases of the roadmaps are Detail (SD) and Validate (DFSS). The primary activities during the *Detail* phase are optimization and documentation. The arrangement, form, dimensions, tolerances, and material properties of all parts are finalized, optimized, and documented. In this respect, however, SD is not dogmatic. It calls for flexibility during all of the phases as activities such as optimization, prototyping, and experimentation could be required at any phase, depending on the nature of the issue. The result of this phase is the specification for production. It is also a time to develop associated manuals and information regarding the use or maintenance of the product.

The *Validate* phase is a time to verify that the designed product meets the specifications in operation as established at the beginning of the project. Emphasis is placed on building a prototype of the product and executing a planned family of tests to validate that the CTS requirements are met. This phase also calls for production installations and subsequent testing and validation of manufactured products to ensure they satisfy the established FRs used in the design process. The final step in this phase is to conduct upfront launch planning, training for production operators, and on-line quality controls to ensure six sigma quality levels.

4 Summary

In this paper we provide a broad look into Systematic Design and Design for Six Sigma, two approaches for designing products. The Systematic Design approach is well structured and documented and has a long history of success in industry. The Design for Six Sigma approach has many different versions and is still evolving. Though Design for Six Sigma addresses the design of products, services, and processes, if we focus at just product design at a high level, both approaches accomplish the same thing: customer driven products. But when one examines the tools and methods used in the major phases of both approaches, there are differences. In this paper we have highlighted those differences. We believe our discussion shows that both Systematic Design and Design for Six Sigma can benefit from an exchange of tools and methods. We did not address, however, exactly how one technique would replace another. Given the history and completeness of Systematic Design, we also did not address the necessity of replacing it's individual tools and methods. Those tasks are left to future studies that are more detailed and narrowly defined. Design for Six Sigma is popular, to a large extend, in that it has been entrained with the implementation of Six Sigma Improvement programs. Given the still dynamic nature of Design for Six Sigma some of the better tools and methods of Systematic Design will no doubt become a part of its future.

REFERENCES

[1] Pahl, G. Beitz, W. *Engineering Design: A Systematic Approach*, 2nd Ed., 1996 (Springer-Verlag, London)

[2] Hubka, V. and Eder, W. *Theory of Technical Systems*, 1988 (Springer-Verlag, Berlin Heidelberg)

[3] VDI Design Handbook 2221: Systematic Approach to the Design of Technical Systems and Products, translated by Wallace, K. VDI-Verlag, 1987.
[4] Pyzdek, T. *The Six Sigma Handbook*, 2003 (MGraw-Hill, NY)
[5] C. M. Creveling, C., Slutsky, J. and Antis, D. *Design for Six Sigma-In technology & Product Development*, 2003 (Prentice Hall, Upper Saddle River, New Jersey, USA)
[6] Yang, K. and El Haik, B. *Design for Six Sigma- A roadmap for product development*, 2003 (McGraw-Hill, New York)
[7] Taguchi, G., Elsayed, E. and Taguchi S. *Robust Engineering*, 2000 (McGraw-Hill, New York)
[8] Altshuller, G. *Creativity as an Exact Science*, 1988 (Gordon and Breach, New York)
[9] Suh, N. *The Principles of Design*, 1990 (Oxford University Press, New York)
[10] Fredriksson, B. 1994, "Holistic Systems Engineering in Product Development," *The Saab-Scania Griffin*, Saab-Scania, AB, Linkoping, Sweden
[11] http://www.qualitydigest.com/aug05/articles/03_article.shtml dt: 11/21/2006
[12] www.isixsigma.com/library/content/c020819a.asp dt: 11/21/2006
[13] Bossert, J. *Quality Function Deployment*, 1991 (ASQ Press, Milwaukee, WI, USA)
[14] Pugh, S. *Total Design: Integrated Methods for Successful Product Engineering*, 1991 (Addison-Wesley, Reading, Mass., USA)
[15] Palady, P. *Failure Modes and Effects Analysis*, 1995 (PT Publications Inc., West Palm Beach, FL, USA)
[16] Hauge, P. and Stauffer, L. "ELK: A Method for Eliciting Knowledge from Customers," *Proceedings of the Fifth ASME Design Theory and Methodology Conference*, 1993, Albuquerque, New Mexico, September13-16.

Contact: Larry A. Stauffer
University of Idaho Boise
College of Engineering
322 E. Front Street
Boise, Idaho 83702
USA
1-208-364-6180 Phone
1-208-364-3160 Fax
stauffer@uidaho.edu