AN ABSTRACT OF THE THESIS OF

Clint W. Peterson for the degree of Master of Science in Mechanical Engineering, presented on April 30, 2007.

Title: Product Innovation for Interdisciplinary Design under Changing Requirements: Mechanical Design for the Bug ID Project.

Abstract approved: ______________________________
Robert K. Paasch

It is crucial for the development of high quality products that design requirements are identified and clarified as early as possible in the design process. In many projects the design requirements and design specifications evolve during the project cycle. Shifting needs of the customer, advancing technology, market considerations and even additional customers can cause the requirements to change. If uncontrolled, design changes derived from shifting requirements may propagate through a design and disrupt the product development schedule, increase development costs, and result in a failure to satisfy the customers' needs. The challenge of designing with changing requirements can be even more challenging in a product development environment where a new product is targeted and/or with interdisciplinary teams. Through work for the Bug ID project, a substantial multidisciplinary project at Oregon State University to generate an automated method for identifying species of particular insects, I have explored possible design strategies for product development under changing requirements. Six design strategies have been generated and implemented in the development of mechanical apparatuses. Based on this experience I offer insights on how to cope with changing requirements in designing a new product, and more important, how to incorporate the considerations of evolving design requirements into feasible product development strategies.

Product Innovation for Interdisciplinary Design under Changing Requirements: Mechanical Design for the Bug ID Project

by
Clint W. Peterson

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirement for the
degree of

Master of Science

Presented April 30, 2007
Commencement June 2007

Master of Science thesis of Clint W. Peterson presented on April 30, 2007.

APPROVED:

Major Professor, representing Mechanical Engineering

Head of the Department of Mechanical Engineering

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Clint W. Peterson, Author

ACKNOWLEDGEMENTS

I would like to thank Dr. Robert K. Paasch for his support and guidance during this project. As my faculty advisor he provided ideas and leadership which contributed greatly to the success of my work. I would also like to thank Dr. Ping Ge and Thomas G. Dietterich for their mentoring in writing a paper published in the International Conference on Engineering Design. I would also like to thank the other members of the Bug ID project who contributed immensely to accomplishments made during my time on the project. I would like to thank Steve Adams and the rest of the Oregon State University Mechanical Engineering machine shop for their work. Additionally I would like thank the National Science Foundation for their financial support of this research.

TABLE OF CONTENTS

LIST OF FIGURES

Product Innovation for Interdisciplinary Design under Changing Requirements: Mechanical Design for the Bug ID Project

1. INTRODUCTION

Every design engineer has a different view of the product development process and for this reason there is no single best design process [1]. Many factors such as size and complexity of the project, the design engineers and customers, and technology changes will influence the product development process. However, following a systematic design process, though it may differ from others, will lead to more successful products [1]. In addition following, in some manner, the widely accepted steps in modern product development should provide a level of experience and insight to any project.

Collaborative product development projects, where there are multiple stakeholders from differing backgrounds, present for design engineers a more challenging work environment. Changing or insufficient customer requirements also provide difficulty in the design process. Both of these issues are becoming increasingly common in today's world of exponentially advancing technology and digitally connected population. Faster-Better-Cheaper, NASA's management strategy for design operations in the twenty-first century reflects the change in world perspective and design interdependency [2].

Despite the challenges they present there are significant benefits to accepting changes to a product being designed as they usually result in an improved product and one that the customer really wants. Changes to a project can represent significant opportunities and can result in a competitive advantage [3]. The design work I did for product development in the Bug ID project presented numerous opportunities to design under changing and insufficient customer requirements. This collaborative project was also the motivation for this research.

In the rest of this section I will review the design process and the issues related to collaborative design. I will also introduce the topic of changing or insufficient requirements, the problems and benefits of change, and the work of this paper to develop guidelines for designing under these conditions.

1.1 THE DESIGN PROCESS

The design process is the set of technical activities that support the product development processes [1]. As products become increasingly complex and design constraints must also meet schedule and budget requirements of an increasingly competitive market, there is a need to continue improvements to the design process and make adjustments to satisfy specific design projects. Understanding the current model of the design process and how design for changing requirements fits into the process will help one understand how the work presented in this paper can better be put to use.

It its most basic form the modern product development process can be broken down into three phases: understanding the opportunity, developing a concept, and implementing the concept [1]. In three words these steps are understand, design, and build. In the understanding stage customer needs are gathered to help define the problem. These needs statements are then analyzed to formulate design requirements which are used to develop the actual design [1]. A problem is understood by comparing the requirements to the design engineers own knowledge [4]. It this regard, every designer will understand the problem differently.

Understanding the design problem, the first step in the design process, is arguably the most important. In product development it is important to develop a set of clear design requirements to help steer the design process in establishing product specifications [5]. It is estimated that poor product definition plays a role in 80 percent of all time-to-market delays. It is also projected that 35 percent of product development delays are a direct result of changes to these definitions through out the design process [4].This problem is usually amplified in multidisciplinary projects where there are many stakeholders from varied backgrounds.

Design requirements start out as qualitative descriptions of what customers need and/or want, extracted from customer survey data, and then are usually transformed into quantitative engineering requirements to facilitate the physical embodiment. There is usually not an obvious relation between the customer needs and the corresponding design specifications for system realization [6]. Since design direction is greatly influenced by these engineering requirements, it is crucial to clarify customer requirements and identify definite design specifications for them as early as possible in a design process.

The second step in the design process is where the traditional engineering takes place. Design requirements must be translated into a design. This process is often approached by decomposing the problem into many smaller sub-problems that are more approachable [4]. These decompositions are generally functional but can also be base on material, position in the design, or any other distinguishing criteria the designer can make use of. These sub-problems are then solved individually and recombined to form a complete design solution [4]. For a given design problem there may be many ideas and even feasible solutions and so an evaluation process is done before a design direction is chosen [4].

The final step involves more than just building the design decided upon. For a manufacturing company this step may be much more involved than the previous two steps as they must now design a means to manufacture and assemble the designed parts. These design processes should be done concurrently with the product design as a good product design does not always make a good design for manufacture and assembly. The field of *design for manufacture and assembly* aims to make parts easier to produce form raw stock and attach together [1].

Another important aspect of implementing the design is modeling. Modeling is done to aid in the embodiment and to validate the design work. The model should demonstrate in a real-world context the functions of the design [1]. Models also assist the design engineers in developing a concept and are often used earlier in the design process. The design process may also result in a prototype that leads to revisions to the design. In this manner the design process is an iterative process that cycles through the steps of design and build outlined above.

I all three of these steps there are opportunities to provide for changes and develop robust designs. The earlier in the design process changes or uncertainties are known the easier they are to accommodate. Thus it is most important to spend the necessary design time to sufficiently gather customer requirements. As the design process progress however, there are many other opportunities to design for changing requirements.

1.2 Collaborative Design

Collaborative design merges the technical and business specialties in an effort the increase product quality and decrease the product development process. Due to the increasing complexity of design problems, and the subsequent specialization to manage advances in technology, large teams of varied engineers often work together through out the design process. Advances in technology are also being applied to more diverse areas forcing engineers to work on design teams with people who may have very little knowledge about engineering principles or the design process. Despite the logistical challenges they present there are significant benefits to designing in interdisciplinary teams.

A design team can be more effective than the sum of its parts [4]. In these situations effectively work together to complete a design project in less time than if the same people were working individually. Project managers use the term synergy to describe these high performing teams [7]. Negative synergy however describes teams who are underachievers. These teams are adversely affected by social, cultural, technological, or physical barriers to communication and understanding. Team communication, trust, and comprehension are crucial to the design process. It is often this lack of understanding that leads to insufficient customer requirements and undesirable changes to the design requirements.

The Bug ID project was a collaborative design project involving multiple mechanical engineers and engineers from other backgrounds as well. The customers in this project were also from varied backgrounds creating a multidisciplinary design environment. This environment was the cause of many of our design changes and helped provide the need for developing guidelines to design under changing requirements.

1.3 Changing or Insufficient Requirements

Design requirements may change for a variety of reasons, and can drastically impact the design of the product [6]. Inadequately defined or changing customer requirements are common problems product design engineers are confronted with through out the design process. A lack of open communication between design engineers and stakeholders is at the root of many of these problems however even with great project communication some changes may be unavoidable.

Customers sometimes may not be clear about what they want, and therefore, their requirements may be underspecified [6]. Even when the customers initially depict their needs, their requirements can change because of cost considerations, advancing technology, or an evolving product development. An existing product may have to change to meet the needs of newly identified customers or market desires. Evolving product development often arises in the development of new products, where a series of prototypes is used in order to learn the nature of a product through several generations: old requirements are tested in the first generation, modified, and then a new set of requirements is identified for the second generation, and so on. Changing design requirements resulting from any of the above factors may affect the course and result of a product design process.

1.4 Problems and Benefits of Changing Requirements

It is important to keep in mind that changing design requirements should be expected during a product design process. Thus it is equivalently important to have an evolving strategy and try to minimize the impact of changes to the design process. Changes in the design process are costly because of the implications are often uncertain, conversely changes can be beneficial in that they may result in an improved product [8]. In project management changing requirements are often called *scope creep* and represent a risk to the project [7].

The general view of scope creep is a negative one. Scope changes are one of the early warning signs of troubled projects and can result in cost overruns, extended schedules, and quality concerns [9]. Requirement changes from poor pre-design or rework is much more expensive than spending the money initially to correctly complete the design phase [4].

Although design changes are usually view negatively, there are times when there can be positive results. Changes to a project can represent significant opportunities and can result in a competitive advantage [3]. Scope creep can be beneficial if it results in a product that better suits the customer. If a customer's needs change then the old requirements may not produce a product that is desirable. In product development it can be useful to work with the client upfront in the requirements phase to define, through a prototyping process, what the client wants. Changes in this process help to better define the final product. Because changes can be beneficial, project managers should work toward managing scope creep instead of insisting on preventing it [5].

In managing scope creep, project mangers must asses the risk associated with a change. Although a change may be beneficial to the design it can be disruptive to the project if it is not simple to implement. Design requirements are often interdependent; the design for one component depends on the design of another component. Changes that will affect several parts of the design are costly in terms of both time and money [6]. The design must be analyzed to determine which changes may be simple to implement and which will significantly impact other parts of the system, often in unpredictable ways. Design engineers can save themselves, and their project, from risky redesign by putting thought into potential changes in the design. In other words, designing for changing requirements can be beneficial to product development.

1.5 Design for Changing Requirements

The work presented in this paper explores possible product development strategies for minimizing the impact of changing design requirements in the context of new product development by interdisciplinary design teams. Through a new mechanical product design done for the Bug ID project [10, 11], I examined effective ways to design when the requirements are not fully defined. As a result I have developed six strategies with satisfying results. In the next section I will review relevant work in the field of design with changing requirements, specifically as related to project management, computer science, and mechanical engineering. In the sections following I introduce the Bug ID project, present these six strategies, and share some concluding remarks about my experiences.

2. PREVIOUS WORK ON DESIGN WITH CHANGING REQUIREMENTS

The possibility of changes to design requirements exists both at the beginning of a new design and for redesign of an existing product. Most product development involves the steady evolution of an initial design. This is often the case to eliminate mistakes through rework and to accommodate new requirements [12]. Although design changes are a normal part of the design process, they are usually expensive and time consuming to implement.

A number of approaches have been developed in academia to assess the impact of a design change [13]. The field of software design has investigated this problem to a much greater extent than that of mechanical design. Luckily, some of the methods developed for software engineering can be applied to mechanical engineering with some adaptation. The difficulty lies in that software design is only concerned with the transmission of information, while mechanical systems must also deal with material and energy transfer.

The field of project management has also had some success dealing with the impact of design changes. Scope creep is managed as a risk to the project, positive or negative [7]. This viewpoint is similar to that of mechanical design engineers and can be translated across disciplines. However, like with software design, project management does not specifically answer the question of how to create designs that are scope-creep-friendly.

2.1 PROJECT MANAGEMENT

Most complex design projects in industry have functional project management in place to help facilitate the design process and organize the interactions of all aspects of the design. Project management in the product development process oversees the non-technical goals of a project and manages the technical aspects to meet these goals [7]. Although project management does not have a direct impact on the particulars of a design they are responsible for incorporating considerations external to the design, into the project. These considerations are most importantly the schedule, budget, and quality of the project [7]. Because all three of these are directly affected by design changes researchers have examined design under changing requirements from the standpoint of a project manager.

Most of the research in project management on the subject of changing requirements has focused on predicting change as a risk assessment tool. Changes to a set of requirements can be costly and difficult to carry out, which makes their impact assessment primarily a monetarily driven objective. Risk assessment can be beneficial to managers in charge of project budget and schedule. Predicting change and change propagation through a product's life span can benefit managers in allocating time and resources [14]. Tracking changes can also give an engineer valuable experience in designing future products.

Many companies have their own methodology or standard process for executing a project. Most start with defining the scope project. The scope may be refined later in the project but it is common practice to establish what work will be done in the project. Included in this phase is the definition and prioritization of both requirements and deliverables [7]. Changes to requirements or deliverables in the form of added design or redesign work are often called scope creep [15]. A clearly defined project scope will help direct and focus a project to its successful completion.

The most common, and often the most expensive, form of scope creep is poor product definition or incomplete design requirements. In the product development process the project scope and the product design requirements are closely tied. Changes to design requirements often result in changes to the project scope. The reason incomplete pre-design work is the most common form of scope creep is that most project managers do not put enough importance to this task [15].

To help mitigate the problem of scope creep there should be a heavy emphasis placed on establishing customer requirements and scope early in the design phase. This may also include spending sufficient time and resources exploring the design problem and alternative solutions early in the design process [15]. The extra time and resources spent on this portion of the design phase can save much more time and money further along in the design process [15].

Project scope management often becomes an iterative process as changes are presented throughout the design process [16]. With any change stakeholders should be kept informed of scope changes that will have an effect on schedule, cost, and quality [16]. Changes should also be tracked throughout the project to reassign resources, and schedules as necessary. A project manager is responsible for managing scope creep, with the goal of trying to minimize the impact of changes to the timeline, cost, or quality of a project [7].

The process of implementing changes to a project should be closely monitored [9]. Have a scope change process to document the requested changes in writing, analyze the impact and cost-benefits, and review changes with the project stakeholders [7]. This process will help to determine the impact of a proposed change before it is accepted. Rules that limit a project's scope can ensure its completion, but a flexible system may recognize more value in a project. Rules should make it difficult for significant changes to occur without discussion and formal approval [17]. However a scope change process should allow changes when necessary.

2.2 Computer Science

In the field of software engineering, changing requirements has been a major problem. Software is often the most flexible and easily changed component of a system [14]. A couple of models used in software engineering consider changes in evolutionary software development. However these models are not appropriate for mechanical design where component interfaces are not as explicit and involve more than just information transmission. These programs generally only identify the immediate implications of change within the immediate sub system and are not capable of exploring the consequences of change propagation through complex systems [18]. Some concepts however, do transfer over to mechanical design.

Perhaps the most important concept that carries over to mechanical engineering is importance of customer involvement in the design process. The socio-technical approach seeks to identify the social, technical, economic, and organizational objectives as perceived by the different stakeholders [19]. This approach contends that communication problems can be reduced if members of the community are involved with all levels of the analysis, design, evaluation and implementation of the system. Design teams with representatives from all the major interest groups have the opportunity to learn from the different viewpoints [19]. However in order for this to be effective users must learn and become familiar with the design methods. Management involved with the design process will become more aware of the possible impact their strategic decisions will have on the system.

Another concept from software development that can be applied to mechanical system design is the importance of knowledge about the interactions between components in a system. The Change Prediction Method (CPM) tool is a software program being developed for predicting change propagation. CPM is a technique for analyzing indirect changes and calculating the combined risk that a change to one component will affect others [18]. Reliable change propagation information is important for successful change management. In complex products, components may be highly interconnected. Changes to one component will likely affect many others including components not directly connected to the component initially being modified. In these products design changes have a major impact on the system in terms of redesign effort and cost. With knowledge about past change propagation, design efforts can be directed toward avoiding change to expensive sub-systems, while allowing change where it is easier to implement [12]. The CPM tool makes use of Design Structure Matrices (DSMs) to provide a simple, compact, and visual representation of the probability that a change will propagate from one component to others [18].

Finally the idea of guessing future changes can be applied to mechanical design. A robust design is one that can cope with alternative futures [19]. In order to build robust systems the designer must attempt to consider all possible alternative futures. The principle outcome of the analysis of the system for future changes is a list of system features which are likely to be affected. *Forecasting horizon* is a term used in the field of computer science to describe the furthest planners can look into the future and conceive of a design to cope with the possible range of requirements at a permissible cost [19]. The greater the forecasting horizon is compared to the target lifespan, the better a designer will be able to conceive of the possible changes and how to account for them. Design engineers have to decide the target lifespan and the decision may be based in some part on their forecasting horizon. Building flexibility into a system is often expensive so it is important to determine where best to build flexibility into the system [19]. The target lifespan will determine how much flexibility the system should have to meet that target.

In his paper, *Adapting to Changing User Requirements*, Land outlines the following guidelines for systems design [19].

- Use design methods that allow prototyping
- Replace the life cycle approach with an evolutionary approach
- Avoid committing to a particular design too early in the design process
- Attempt to identify aspects of a design which are volatile and subject to change
- Build flexibility into the design

He notes that these guidelines may not be applicable to all design problems but the designer can choose which will be most effective for the current application. For the application of mechanical design they are quite relevant.

2.3 Mechanical Engineering

The importance of the early phases of the design process and specifically, careful requirements development is stressed in the field of mechanical design as well as computer science and project management. A clear set of design specifications is evidence the design team understands the problem. Ullman states that design requirements should have the following qualities;

- Discriminatory
- Measurable
- Independent
- Impartial

These ensure that the set of requirements reveal the differences between design alternatives without allowing ambiguity or dictating a design [4].

Analyzing requirements is an important step in the product development process. It is the last step in the gathering qualitative customer requirements phase, and can be very beneficial in the next phase, generating quantitative engineering requirements. According to Yoo, Catanio, Paul, and Bieber, “the analysis phase of the systems development life cycle strives to precisely and comprehensively isolate and understand the problem domain and to document what is to be built.” [20] This step is often done a number of times through out the design process when an iterative design approach is used.

Analyzing requirements is the process of evaluating and organizing a set of requirements [1]. It is done to assume completeness of the requirements gathering phase and ease the transition to engineering requirements generation. It may be beneficial to categorize a list of requirements into component groups. Breaking requirements down into a structured set of component requirements can make the requirements specification more amenable to analysis. This will not guarantee completeness and correctness of the existing requirements, but it will serve to increase confidence in such specifications by identifying inconsistencies

Customer requirements can be broken down into a functional structure, to abstractly represent a product and its customer needs in terms of function instead of components [1]. The function of a product is a statement of the relationship between the input and output of a product [1]. Product function states what the product is to do. A design problem can be decomposed into sub-functions which when completed, satisfy the overall function [1]. Functional decomposition is useful in analyzing customer requirements because it is abstract and can be represented without a specific model. In other words abstraction ignores what is particular and focuses on what is essential, the function of the product [1]. It is necessary to examine the relationships between sub-functions in order to understand the compete design. This can be useful in determining how a product should link, connect, or transform its inputs to outputs. In this practice one should consider the design interaction of geometry, material, energy, and information [1]. It is the interaction of all of these that make mechanical design so challenging.

Some customer constraints cannot be satisfied by a function however and must be considered differently. Criteria of cost, weight, size, reliability, and “looks” are examples of non-functional based requirements [1]. These requirements can be called human factors requirements [4]. No matter how the requirements are organized or classified they must then be translated into a design. Unfortunately little has been researched in this next step for designing with changing requirements.

Freezing requirements is one way design engineers try deal with changing requirements. One goal of a freeze is to reduce the likelihood design changes. The major benefits from using design freezes are the ability to structure the design process and to control design changes [13]. A design freeze marks the end of a development stage where requirements become fixed before the design can continue [13]. Early design freezes have the benefit of pushing any design changes to future product generations. This can be constructive in an iterative design process. Early design freezes however, can also force a design before it is beneficial to do so. When the exact product requirements are uncertain, it may be advantageous to postpone a design freeze. Some changes due to safety concerns, problem corrections, or altered customer requests will still have to be carried out regardless of whether a requirement is frozen. Changes after a freeze are likely to be more costly, and the cost will continue to increase the later the change is implemented [13].

Many design engineers feel it is best to keep parts flexible where changes are anticipated. In stead of having a single freeze date for the design early in the life cycle, some aspects of the design can remain open for re-design with further information [19]. This enables the design to more closely meet the requirements at the time of implementation. An ongoing process of analysis and evaluation carried out in parallel with design and construction [19]. Information about design freezes is especially important when working in a design team. Recognizing the dependencies between parts and the acknowledging which parts may be frozen can avoid inadvertent changes to the overall design.

The Design For Variety (DFV) method uses product platform architecture to provide a structured approach to reduce the amount of redesign effort for future generations of a product. For large projects, a system architecture can be used to break down the design into smaller subsystems at each level of the design hierarchy [6]. The DFV method has the advantage of being a simple and inexpensive technique to determine potential design changes. The methodology makes use of standardization and modularization techniques to reduce future design costs and efforts [21]. The design for variety method develops two indices to measure a product's architecture. The first, called the Generational Variety Index (GVI), is an indicator of the amount of redesign effort required for future iterations of a product. The other is called the Coupling Index (CI), and it is used to gauge the extent of coupling among the different components in a product. DFV can be used to help reduce the impact of variety on the life-cycle costs of a product [21].

Overall little literature is available that specifically addresses the problem of designing mechanical devices with changing or incomplete customer requirement. Placing an emphasis on the requirements gathering phase of the design process is a widely excepted practice for any design team and is applicable to this situation as well. The translation of customer requirements into a design is vulnerable to changes and techniques for designing under these circumstances would be beneficial to mechanical design.

3. BUG ID PROJECT BACKGROUND

The Bug ID Project is a collaborative research development effort by a multidisciplinary team of entomologists, computer scientists, electrical engineers and mechanical engineers, to generate an automated mechanism for identifying species of particular insects [11]. The Bug ID project seeks to advance ecological monitoring through automated identification of insects using machine learning and pattern matching techniques [10]. By coupling computer algorithms, mechanical manipulation, and high-resolution photographs, it is the hope that extensive insect population counts can be obtained inexpensively. Such information would be invaluable to ecological science and environmental monitoring.

Among several teams involved, the mechanical engineering team is responsible for designing and building a mechanical apparatus to capture images. The mechanical design requirements for the project are centered on providing quality images for the computer science team to develop and test their identification algorithms. Hence, the computer science engineers are the primary customers for the mechanical design work. There are also requirements from the entomology team in terms of dealing with insects preserved in glycol. Finally, because the mechanical engineers also were tasked with performing specimen manipulation and photography, we are our own customers, with usability and functionality requirements.

Our general design approach is an iterative process with the evolution of changing design requirements through several generations. At the very beginning, the project focused on stonefly larvae, an indicator species for water quality in streams and rivers. These insects range from a centimeter to over five centimeters in length and are most easily distinguishable by the patterns on their backs [10]. As progress was being made in creating identification algorithms by the computer science team, work was started on a different group of insects known as soil mesofauna. These tiny organisms live in soils and are sensitive to soil type, chemicals in the soil, and land management procedures, making them excellent indicators of soil biodiversity [10]. Images of these two groups of insects can be seen in Figure 1. The two groups of insects are physically different enough to have unique requirements for image quality and to require separate mechanical apparatus for capturing images.

Figure 1 - Stonefly larvae on the left and soil mesofauna on the right

Targeting the above design challenge, the mechanical design team has developed a working prototype for identifying stonefly larvae and soil mesofauna, shown in Figure 2. The basic design consists of an insect holding and viewing apparatus, two separate transport mechanism, and a camera in conjunction with a microscope. For the stonefly larvae, a transport tube guides the specimen from one holding bin to another and into position under the microscope. The larvae are kept in glycol at all times, and they are transported by fluid motion with pumps. Half of the tube is a blue plastic to provide a blue background in the images. This is useful for image segmentation. The apparatus is placed under the microscope.

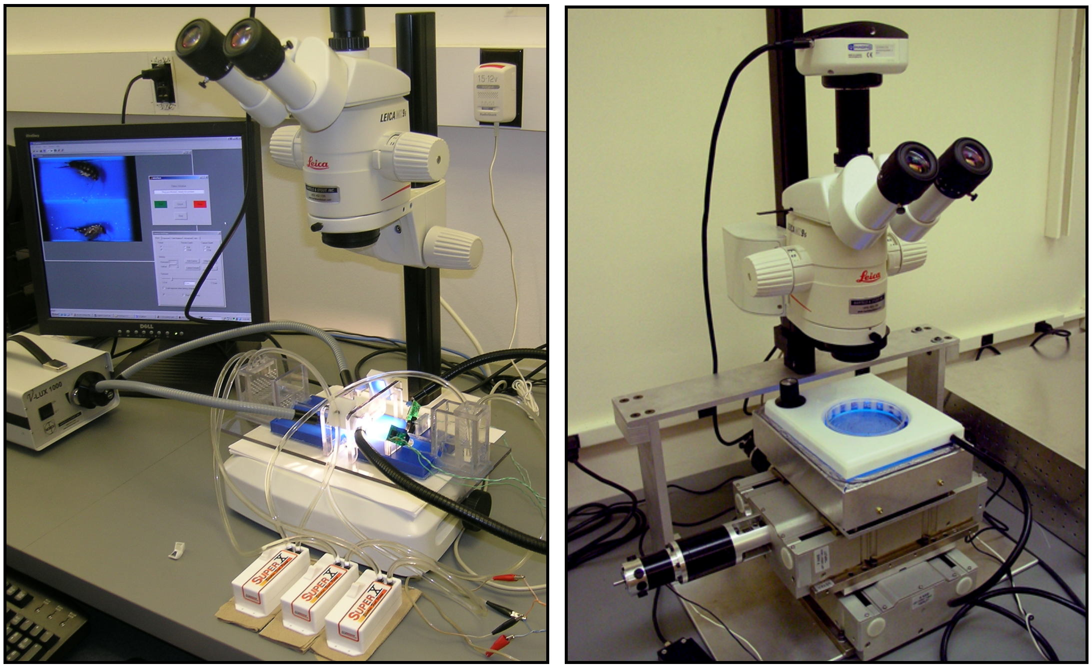

Figure 2 - Stonefly larvae apparatus left, soil mesofauna apparatus on the right

For the soil mesofauna, the holding and viewing apparatus was simply a Petri dish sitting on a horizontal LCD screen. The LCD screen is used to give the images a blue, green, or red background for segmentation and transparency estimation. The screen is mounted on top of two motorized stages, for motion in the *x* and *y* directions. All of this was then bolted to a platform that held the microscope column above the LCD screen. Images are obtained using a high resolution camera mounted on the microscope. The camera is controlled by a computer from which one can view the images in near real time. The design also calls for a way to extract identified organisms from the Petri dish, possibly via a pipette attached to a robot arm.

The ultimate goal of the project is to have a new product that meets all of our customers' needs, but even at this stage, all the specifications are still not known [11]. From the beginning of this project, the mechanical design team has been working under a very loosely-defined set of customer requirements given by the computer science team who are concurrently developing their recognition algorithms. Due to the nature of high-level design uncertainty in developing a new product with partners from other disciplines, the design requirements could not be fully defined up front and had to be explored through a process of prototyping and testing. This has rendered the product design process highly unpredictable and brought about constant changes to both the design requirements and the product design. Throughout the four years of this project, there have been multiple product development iterations and the design is still being improved.

With the above working prototypes for identifying stonefly larvae and soil mesofauna, design improvements and alterations continue to necessitate changes by the mechanical design team. In order to ensure the quality of the apparatus and the success of the Bug ID project as a whole, we must have some effective ways to enable us to keep up with the changing requirements, and moreover, minimize the impact of the changes to the existing design. This new product development experience, with partners from other disciplines, has greatly motivated us to develop the six strategies presented in Section 4. Though the Bug ID project is a substantial undertaking, it entails a relatively small-scaled mechanical system design. The challenge of dealing with changing customer requirements, however, is representative for any new product design, particularly when multi-disciplinary teams are participating, and the presented strategies are expected to be transferable to cases involving larger, more complex systems.

4. PRODUCT DEVELOPMENT DESIGN STRATEGIES WITH CHANGING REQUIREMENTS

In light of managing changes in design, based on our experiences with the new product development in the Bug ID project, we have developed and implemented the following six strategies described in the subsequent sections; 4.1–6. The goal is to share our experience with others, in the hope that the presented strategies can facilitate a successful product design in a similar situation.

4.1 Establish and Foster Open Communication between Design engineers and Customers

Coupling active customer participation with rapid prototyping methods is the foundation for identifying unanticipated customer requirements in a timely fashion [6]. Design engineers face numerous obstacles that may hinder a product design. Errors in designs often are the result of miscommunication between domains, rather than within the domains where design engineers are experts [22]. Effective communication between the design engineers and customers is important throughout the product development process to help prevent problems.

It is important to have an effective interface with the customers during the entire design process. This interface is responsible for exchanging important information regarding changes to design specifications [6]. To help facilitate the interface between the customer and the design team, it is important to make communication as open and as simple as possible. There must be an understanding of the potential barriers and impediments to communication between involved parties.

Real impediments to communication may exist for which there may not be any simple solution. In our example project, two of the computer science engineers working on the Bug ID project are at a different university and rarely communicate face to face with the rest of the group. Communication was primarily through phone conferences and email. This is a physical barrier but one that can be mitigated by recognizing the limitation and directing communication through other means.

There may also be perceived barriers that emerge from the history and culture of the group and the individuals involved [6]. Junior level project members may feel intimidated or lack the confidence to speak their ideas. An environment that is open and encouraging can help these individuals contribute to the project. Fostering open communication requires strong social relationships built on trust within the design team and between customers and developers. Every design team will have its own perceived barriers that may be best dealt with in their own way. However making an effort to seek out these barriers and alleviate their effect will prove beneficial to the design team.

Supporting an environment of open communication was used in the Bug ID project among the interdisciplinary partners. Communication was encouraged through email to all members and biweekly meetings where requirements were implicitly or explicitly a topic. This allowed everyone a chance to speak their requirements and discuss design features. Because our principle customers for the mechanical design were other research partners on the project as they all have a vested interest in the success of the design, this approach was particularly effective in enhancing communication and developing requirements. This may not be typical in product development projects but can always be fostered through open communication channels. Going through the product assessment process together, a customer and design engineer can help create a better understanding of product direction and design requirements.

4.2 Generate a Complete List of Requirements and formally analyze them

One important aspect of the design process and change management is the development of customer requirements [4]. Customer needs are expressed as written statements that are developed by interpreting the information gathered from customers [5]. An up-to-date list of requirements should be explicitly written down. A complete list may never be realized as it is often growing or changing through out the design process. The list should account for all the customers, interfaces between components and functions, and include estimations where complete specifications are not known.

A list of requirements is especially important for iterative designs and product evolution because a complete list of requirements is used for future evaluation. When changes are made to requirements, a new revised list must be formulated. Separate and iteration-specific lists are useful for tracking evolutionary changes in the original requirements. They will also help with prototype evaluation and to identify conflicts through out the design process.

The up-to-date list should then be analyzed for completeness and seek out missing requirements. It is helpful to decompose requirements into component parts and structure the requirements hierarchically so they are easy to follow and analyze. It is also important to identify and define inter-requirement rules and relationships between different components and functions [3]. These may be discovered only after examining the gathered requirements or after the design work has started.

Some requirements may be unspecified; either they are considered to be relatively unimportant or they are assumed to be intuitive and are never explicitly stated. It is the responsibility of the designer to consider all of the requirements for the design even if they are unstated or perceived as unimportant [6]. At times customers do not know what they want, and requirements may be underspecified. Customers or design engineers must then make educated initial guesses as to what their needs will be. If a design is still not sufficiently specified, the designer may have to make estimations about engineering requirements. A successful designer is able to interpret what the customer really needs when adequate, explicit requirements are not given. This may be facilitated by observing design trends and making predictions, or by creating customer needs where there were none before. Both of these approaches can be successful if the customers are well understood and there is sufficient communication [6], as stated in Section 4.1. As more knowledge is gained, the requirements may be refined and properly specified during the product development cycle.

Requirements must also be created for interfaces between components or functions. These are then used to check consistency, track change propagation, and as a measure of completeness. These are often implicit assumptions, but they should be explicitly stated to clearly show the flow of information, energy, and material in the overall system [3]. It may be beneficial to categorize a list of requirements into component groups. Breaking requirements down into a structured set of component requirements can make the requirements specification more amenable to analysis. This will not guarantee completeness or correctness of the existing requirements, but it will serve to increase confidence in such specifications by identifying inconsistencies [3].

There were issues in the Bug ID project with trying to match the Bug ID program with control for the *x*-*y* stage. The firmware commands arbitral chosen by the mechanical team were not the same written into the software and had to be changed. Initial stipulation for these components would have troubleshooting time and effort in coupling these two components. It can be easy to put aside requirements that do not contribute directly to the function of the product, but they are important to the product development process.

The design must account for all the customers of the product [6]. Often in product development the initial end users are the design engineers, as they test the product for future improvements. Customers may also be other design engineers who are responsible for designing coupled components. Anyway to address each individual stakeholder will be beneficial to the product. Although in the Bug ID project our customers were readily available there were times when a member who missed an earlier meeting would request a change to meet a requirement. An email requesting input after the missed meeting might have avoided the need for making a change to design work that was already done.

One obstacle to avoid when developing design requirements is over specification. It is easy for requirements to become over specified, which results in unnecessary functionality and complexity. If it is unclear if a requirement is really necessary, it may be beneficial to test it separately. For the Bug ID project, the design team tested the importance of different color backgrounds by placing transparent colored plastic sheets under the microscope. It was agreed that the requirement was significant, and it was eventually achieved with an LCD screen component.

4.3 Distinguish between What Won't Change and What Will

When designing with changing requirements, it is important to identify and classify current customer requirements. Categorize them into enduring and provisional engineering requirements. Early identification of requirements, functions and architectures that are stable and of requirements, functions and architectures that are subject to evolution, can provide stability to early states of development. Some components will be more fundamental and should be given more attention.

Recognizing which requirements are likely to remain in future iterations will direct the direction of design.

Requirements can be examined from a qualitative perspective. Sometimes, the changing requirements identified are parametric: the qualitative requirement is known but the quantitative target may migrate during development. In other words, a requirement may not be strictly an enduring or a provisional requirement. It is more likely that there will be a continuum and that requirements should be rated on their degree of stability. With rated requirements in hand, one can proceed with a design direction focusing the majority of attention on enduring components. Other components that are likely to transform, can be targeted differently than more permanent ones, in order to minimize some of the negative affects of design changes.

This approach was used for the Bug ID project. A base platform was developed first for the soil mesofauna apparatus. This mounted the microscope above an *x-y* stage. This satisfied the enduring requirement of fixing the microscope with respect to the motion of the sample. We could then test different components for stage translation, background color, and additional lighting. The base platform was designed to be robust and flexible to accommodate future components. It has not been altered in several design iterations. Figure 3 shows the components of the soil mesofauna design.

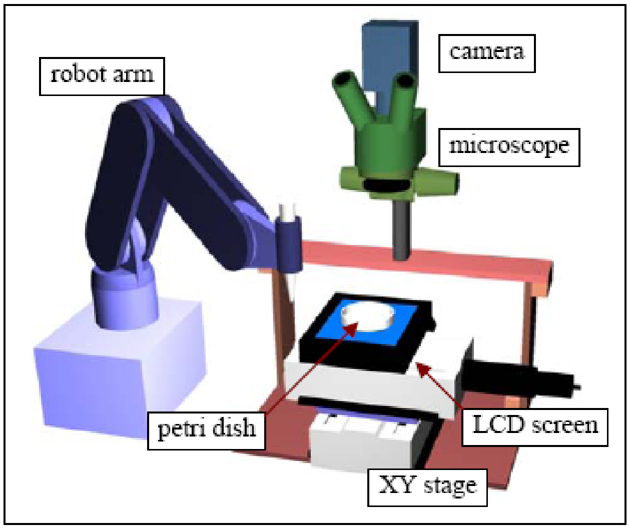

Figure 3 - Components of the soil mesofauna apparatus

Another way to find out what won't change is to freeze certain requirements. This can reduce the risk associated with changes by only allowing only some of the requirements to change. Ideally, components that are highly connected and are expensive from a budgetary stand point, should have a higher priority for being frozen. In an iterative design process, freezes can establish preliminary information as the basis for future design work. Off-the-shelf parts are already frozen and can provide a starting point for the design process. They can also reduce the risk in performance and design and also reduce the workload of the designer [13]. Design freeze, however, does not guarantee that a requirement will not be changed, and it should only be considered when there is strong communication between customers and design engineers.

When starting design work for the soil mesofauna part of the Bug ID project it was decided early on that the same camera and microscope would be used. This froze these components in the design and allowed us to focus on adapting other components to meet all requirements of the design.

4.4 Predict the Future

Future Analysis, a method used in software engineering, offers techniques that may aid early identification of requirements subject to change [19]. Prediction of changes in the design process should be analyzed continuously throughout the design process and in each of the sub-groups involved in the project. Customers or partners in the project may be able to forecast possible requirement changes. For example, one may have new information about their design work and how their needs may change. Another common situation is designing a prototype with future plans to redesign based on assessment of the prototype. Knowledge of this sort can help place priorities on engineering requirements. It is beneficial to identify the potential for change as early as possible in the design process. Experience and past history on the project can be influential in predicting changes [8].

In our Bug ID project, it was known early on that a blue background would be necessary for the images, however it was not known how bright or how blue the background would need to be. There was also talk of using a green or red background in addition to the blue. An LCD screen was chosen because it was versatile in terms of color and brightness. This initial prototype allowed testing of the design space in terms of color and brightness. Another component in the project is a robot arm used extract mesofauna once identified. This has not been implemented into the current prototype at this point, but allowances have been made to accommodate the component once the design is finalized.

4.5 Utilize an Iterative Product Development Strategy

One strategy to deal with changing design requirements is to adopt an iterative production development strategy with an emphasis on quickly producing designs that meet current requirements. Sometimes customers do not know what they want, or do not want, in a product until they see a physical prototype. It can also be the case that customer's wants change after seeing and using a prototype. Our general design approach in the Bug ID project (see Section 3) was an iterative approach, using prototypes to test and formulate requirements. Design requirements and product architectures are developed iteratively, simultaneously addressing requirements specification and product design.

To be effective, this strategy requires quick turnover of designs and prototypes. Evaluation and frequent prototyping can help identify conflicts throughout the design process [22]. The iterative design approach may forgo quality for quickness with the understanding that the current design will be altered and need to only serve the purpose of advancing the design process. This approach may be unavoidable in situations where testing is required to absolutely define customer requirements.

The iterative strategy still requires examination of the customer requirements. It is imperative for each iteration to have a definite set of requirements in order to assess the current model. Thus, each iteration is a complete design process consisting of the following steps: identify customer requirements, formulate engineering requirements, produce a model, test the model, and re-examine engineering requirements. Requirements for the new model can be compared to the requirements for the existing model so that the success of the current model can be applied to the next model.

Often in design of new products, all of the customer requirements are not known or are not concrete. Only through testing and further information about the product can some requirements be worked out. Thus it is important to work closely with customers to help develop these requirements. The assessment process can be accelerated with rapid iterations of prototypes. A trade off decision to make is how much time and effort to put into each iteration. Higher quality iterations may be more informative, but a number of rapid iterations can test a broader range of components. Early on, product quality can be substituted for product quantity. Then as the design evolves into something a little more concrete, the focus should switch to higher quality iterations of the product.

It may be beneficial in an iterative approach to hold some design components the same from model to model (as described in section 4.3). By changing a single or small number of components, it is easier to test and identify the success of those particular components. This practice can be facilitated by identifying those requirements and corresponding components that are more enduring and center design work on those components. Once a quality model foundation has been established, further components can be implemented and tested.

In the Bug ID project, when we started the design for the stonefly larvae apparatus, several iterations of the transport tube were tested. Different sizes, shapes, and lengths of tubing as well as special groves cut in solid plastic were part of the iterative process. The transport contraption was designed, built, and tested several times before a design was selected and incorporated into the larger apparatus. Since then, there have been several more iterations of the transport contraption, shown in Figure 4, to alter the viewing angle and direction. All of this information will be useful when we decide to move on to the next iteration.

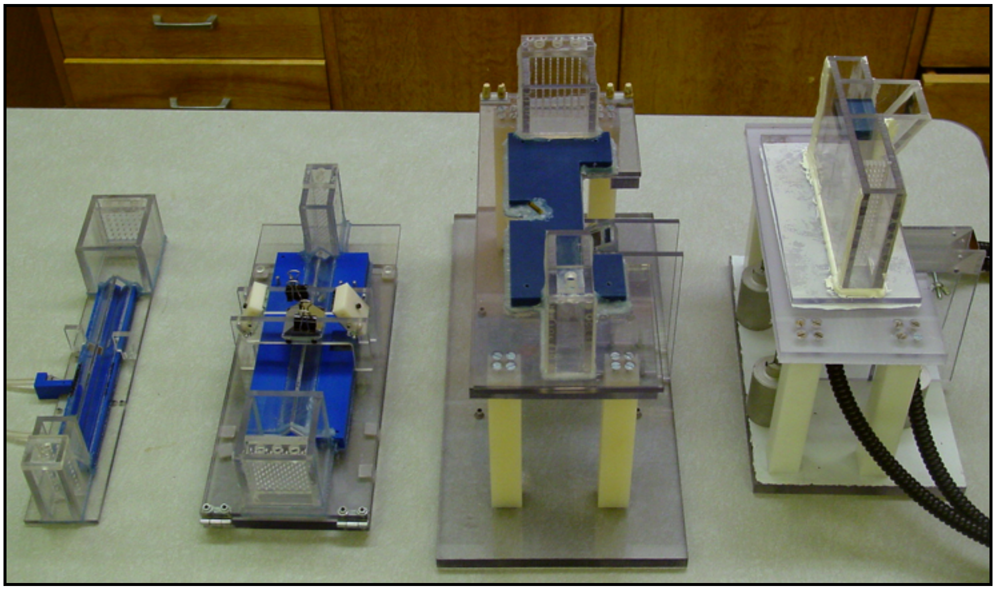

Figure 4 - Several iterations of the stonefly larvae transport mechanism shown chronologically from left to right

An iterative approach was also used for the design of ring lighting for the soil mesofauna apparatus. Three of the designs are shown in Figure 5 below. The first prototype tested the concept of lighting from the side. The prototype proved that side lighting provided shadow free illumination of the specimens, but it was also determined that more light was required. The second prototype tested new brighter LEDs and manufacturing by CNC mill. Finally the third prototype provided even more light, simplified the wiring, attached to the LCD screen, and located the Petri dish in the center of the ring. The iterative process resulted in a satisfactory design however the same result might have been achieved with less work by selecting product architectures that tolerate changes to the design.

Figure 5 - Iterations of ring lighting shown chronologically from left to right

4.6 SELECT PRODUCT ARCHITECTURES THAT TOLERATE CHANGING REQUIREMENTS

Requirement changes are likely to occur in long-lived systems. Design engineers can design for change by embedding flexibility into a design, instead of resisting change or passively accepting it. More and more design engineers are realizing that their designs are expected to have longer life-spans in an increasingly complex and dynamically changing environment [23]. Designing with changing requirements in mind, it is often advantageous to design components to be flexible, allowing them to accommodate a wider range of possible requirements. Incorporating adaptable product architectures is one way to design for changing requirements.

Product lines and product families that provide the core requirements while allowing other requirements to change, build flexibility into a design. Product architectures are effectively layouts of components and subsystems [1]. The focus of product architectures is to transform the product function into product form [1]. A classic example of flexible product architecture is the Erector Set. A countless number of products can be made from the metal beams with regular holes for nuts, bolts, screws, and other mechanical parts. Other more practical examples include standard bolts and nuts, hose and hose fittings, gear sets, chains, and joinable materials. In the field of concurrent product development, design engineers create their own flexible architectures by designing a multitude of product from the same parts. For example cordless power tools often use the same battery, motor, housing materials, and periphery attachments.

Flexible components may be called over-designed. Over-design is not optimal in design work, but it can prove cost effective for future generations of a product. Over-designed prototypes can be used to quantify a qualified requirement. In the design iteration of the ring lighting for the Bug ID project, the second prototype provided much more light than the previous prototype and the ability to adjust the brightness. This was done in an effort to quantify the quality requirement of *bright LED lighting*.

Different requirements and corresponding design features can create the need for tradeoffs in a design. However, simply increasing the design flexibility also increases its complexity, which in principle is undesirable. A designer should not create what can be designed simply to promote flexibility; rather it is important to determine what should be designed [6]. An assessment will have to be made whether the benefits of flexibility outweigh the cost to functionality.

Specifications should also be analyzed to determine if they are likely to change. Requirements that are likely to change should be translated into flexible components as long as functionality is not adversely affected. In analyzing the requirements, it may be found that flexibility can be incorporated into some components, but others may then find it hard to achieve their desired function. For those requirements that are not likely to change, more emphasis should be placed on functionality, because these components are likely to carry on throughout design changes.

In the designing the light rings for the soil mesofauna apparatus it would have been beneficial to use a LED architecture that allowed for changes. A system to attach more LEDs or different LEDs may have eliminated the need for one of the design iterations. Instead the LEDs were soldered together at pre-drilled holes in the housing. The power supply for each design however was the same and the product architecture for the electrical connection allowed for testing each design without changing the apparatus. Flexible product architectures may be difficult to utilize but in projects with changing requirement the benefits are often worth the extra effort.

5. CONCLUSIONS

Design direction is driven by customer requirements and engineering specifications. It is crucial for producing quality designs within budgetary constraints to identify and stabilize these requirements as early as possible in the design process. Under ideal conditions, the requirements and specifications are invariant, and the design proceeds in a sequential manner to a final product. In many projects, however, the requirements evolve during the project. These changes are often responsible for disrupting the product development schedule, increasing development costs, and failing to meet requirements.

Changing customer requirements can come about for a number of reasons and can occur at all stages of the design process. In an interdisciplinary design project, partners from different disciplines must work collaboratively to deliver a desired product design that will satisfy evolving requirements of everyone involved. In these situations, product requirements especially are interdependent; specifications from one component may depend on or restrict specifications for another component. Thus a change to one requirement can propagate through a product and cause numerous other changes. The issue of designing under inadequately defined requirements in new product development can have similar consequences.

Current design strategies try to limit changes to design because of the negative effect they often have on a project. This approach however is reactive and only addresses the problem of changing requirements after they have occurred and it is too late to adjust the current design. Design approach should be proactive, designing flexibility into systems and planning for changes. Designing with changing requirements in mind can be especially effective when a customer has only loosely identified requirements or when requirements are not fully known.

Through the development successes and failures of the Bug ID project, our design team gained valuable insight into the process of designing under changing requirements. With the project's evolving requirements, we have developed six product development strategies to cope with changing requirements and specifications. They were tested while developing working product prototypes for the project. Through this paper we would like to share our experience and offer these six recommendations for designing with changing requirements.

1. Establish and foster open communication between customers and design engineers. This includes communication within a design team.
2. Develop and explicitly write down design requirements as soon as possible. It is important to identify requirements for component interfaces and other possible unspoken product specifications. Analyze the list for completeness and to seek out missing requirements.

3. Examine the list of requirements to identify which requirements are likely to change and which are stable. In the early stages of design spend more time on the enduring components.
4. Predict future customer needs and requirement changes. Make allowances for changes and create flexibility in components to accommodate future changes.
5. Use an iterative approach to product development. Quick turnover of designs and prototypes provides a method for testing requirements and discovering unanticipated requirements.
6. Build flexibility into a design by selecting product architectures that tolerate changing requirements. This can be achieved by over-designing components to meet future needs, particularly in components that are likely to change.

These guidelines are intended to reduce the negative impact of requirement changes to the product development process. Although the Bug ID project was a relatively simple design effort the concepts should apply to any design process. However every idea present in this paper will not be applicable to every project. Each project will be different and design engineers will have to apply what their project allows to be beneficial. Successful design engineers are able to utilize a design process to produce a solution to the problem at hand. The more knowledge one has about the design process the better the process will match the problem and the more successful the solution.

6. REFERENCES

[1] Otto, K. and Wood, K. *Product Design*. 2001. (Prentice Hall PTR).

[2] Paté-Cornel1, E. and Dillon R., 1998. *Analytical Tools for the Management of Faster-Better-Cheaper Space Missions*, IEEE Aerospace Conference Proceedings, New York, NY, 5, 510-530.

[3] Russo A., Nuseibeh B. and Kramer J. *Restructuring requirements Specifications for Managing Inconsistency and Change: A Case Study*. In Proceedings of 3rd International Conference on Requirements Engineering ICRE `98, Colorado Springs, April 1998. pp. 51-61.

[4] Ullman D.G. *The Mechanical Design Process, 3rd Ed.* 2002 (McGraw-Hill, New York).

[5] Ulrich K.T. and Eppinger S.D. *Product Design and Development, 2nd Edition*. 2000 (McGraw-Hill).

[6] Hintersteiner J.D. *Addressing Changing Customer Needs by Adapting Design Requirements*. In First International Conference on Axiomatic Design, ICAD2000, Cambridge, MA, June 2000, pp. 290-299.

[7] Gray, C.F. and Larson, E.W. *Project management: The managerial process,. 3rd Ed*., 2006. (McGraw-Hill).

[8] Strens M.R. and Sugden R.C. *Change Analysis: A step towards Meeting the Challenge of Changing Requirements.* In IEEE Symposium and Workshop on Engineering of Computer Based Systems, ECBS '96, Friedrichshafen, March 1996, pp. 278-283.

[9] Giegerich, D.B. *Early Warning Signs of Troubled Projects*. AACE International Transactions, 2002, p02.1, 8p.

[10] Mortensen E.N., Delgado E. L., Deng H., Lytle D., Moldenke A., Paasch R., Shapiro L., Wu P., Zhang W., and Dietterich T.G. *Pattern Recognition for Ecological Science and Environmental Monitoring: An Initial Report*. In N. MacLeod and M. O'Neill (Eds.) Algorithmic Approaches to the Identification Problem in Systematics. (in press).

[11] Zhang W., Deng H., Dietterich T.G., and Mortensen E.N. *A hierarchical object recognition system based on multiscale principal curvature regions*. International Conference of Pattern Recognition, 2006, 1475-1490.

[12] Clarkson J.P., Simons C., and Eckert C. *Predicting Change Propagation in Complex Design*. ASME Journal of Mechanical Design, 2004, 126(5), 765-797.

[13] Eger T., Eckert C., and Clarkson J.P. *Restructuring The Role of Design Freeze in Product Development.* In 15th International Conference on Engineering Design, ICED `05, Melbourne, August 2005.

[14] O'Neal J.S. and Carver D.L. *Analyzing the Impact of Changing Requirements*. In 17th IEEE International Conference on Software Maintenance, ICSM '01, Florence, November 2001, pp. 190-195.

[15] Kuprenas, J.A. and Nasr, E.B. *Controlling Design-Phase Scope Creep*. AACE International Transactions, Morgantown, 2003, CSC 01.1-5.

[16] Khan, A. *Project Scope Management*. Cost Engineering, 2006, 48(6), 12-16.

[17] Gary, L. *Will Project Creep Cost You - or Create Value?* Harvard Management Update, 2005, 10(1), 3-5.

[18] Keller R., Eger T., Eckert C., and Clarkson J.P. *Visualizing Change Propagation*. In 15th International Conference on Engineering Design, ICED `05, Melbourne, August 2005.

[19] Land, F. *Adapting to Changing User Requirements*, Information and Management (5) 1982 p. 59-75.

[20] Yoo, J., Catanio, J., Paul, R., and Bieber, M. *Relationship Analysis in Requirements Engineering*. In Requirements Engineering, 2004.

[21] Martin M.V. and Ishii K. *Design for variety: developing standardized and modularized product platform architectures.* Research in Engineering Design, 2002, 13, 213-235.

[22] Odell, D. and Wright, P. *Concurrent Product Design: A Case Study on the Pico Radio Test Bed*. Masters Thesis, University of California at Berkeley. 2002.

[23] Jordan N.C., Saleh J.H., and Newman D.J. *The extravehicular mobility unit: A review of requirements, environment, and design changes in the US spacesuit*. Acta Astronautica, 2006, 59(12), pp.1135-1145.