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

PRODUCT INNOVATION FOR INTERDISCIPLINARY DESIGN UNDER CHANGING REQUIREMENTS

Clint Peterson[1] Robert K. Paasch[2], Ping Ge[3] and Thomas G. Dietterich[4]

[1]Oregon State University, Department of Mechanical Engineering
[2]Oregon State University, Department of Mechanical Engineering
[3]Oregon State University, Department of Mechanical Engineering
[4]Oregon State University, School of Electrical Engineering and Computer Science

ABSTRACT
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. Because the different parts of a product are usually inter-connected, the requirements and specifications for one part of a product are often dependent on the requirements and the evolving design of other parts of the product. If uncontrolled, the design changes derived from evolving 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. The work of this paper explores possible design strategies for product development under changing requirements. Six design strategies were identified and implemented in the development of an insect imaging device for the Bug ID project at Oregon State University. Based on experiences throughout this interdisciplinary project, this paper explores how to incorporate the considerations of evolving design requirements into feasible product development strategies.

Keywords: Changing requirements, design strategies, product innovation, interdisciplinary teams

1 INTRODUCTION
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. Many factors such as size and complexity of the project, the design engineers and customers, and technology changes will influence the product development process. Inadequately defined or changing customer requirements are common problems design engineers are confronted with in product development projects, however the present design process literature offers few amendments for working in this environment.

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]. In product development it is important to develop a set of clear design requirements to help steer the design process in establishing product specifications [2]. 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 [3]. Thus understanding the technical design problem as well as how it fits into the context of customer needs is crucial to for a successful design process.

Design requirements may change for a variety of reasons, and could drastically impact the design of the product [4]. Customers sometimes may not be clear about what they want, and therefore, their requirements may be underspecified [4]. 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 new 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.

Changes in the design process are usually costly because the implications are often uncertain, conversely changes are also often beneficial in that they may result in an improved product [5]. Changes to a project can represent significant opportunities and can result in a competitive advantage [6]. 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 increasingly dramatic in terms of both time and money [4]. Requirement changes from poor pre-design or rework is much more expensive than applying the resources needed initially to correctly complete the design phase [3]. 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 a design strategy that actively attempts to minimize the impact of changes to a developing design. Design engineers can save themselves, and their project, from risky redesign by putting thought into potential changes in the design.

Collaborative product development projects, where there are multiple stakeholders from differing backgrounds, also present for design engineers a more challenging work environment. 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 and they have become common place in many engineering industries.

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 [7, 8],the mechanical team examined effective ways to design when the requirements are not fully defined and evolve as the design progresses. As a result have six strategies were identified and applied with satisfying results. In the following sections the Bug ID project is introduced, a review relevant work is given, and these six strategies are presented with examples from the Bug ID project.

2 DESIGN WITH CHANGING REQUIREMENTS IN THE BUG ID PROJECT

The Bug ID Project is a collaborative research development effort by multidisciplinary teams of entomologists, computer scientists, and mechanical engineers, to generate an automated mechanism for identifying species of particular insects [8]. The Bug ID project seeks to advance ecological monitoring through automated identification of insects using machine learning and pattern matching techniques [7]. 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.

The customers in this project are from varied backgrounds creating a multidisciplinary design environment. This environment was the cause of many design changes and helped provide the need for developing guidelines to design under changing requirements. 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 and with device usability and functionality requirements. In the long term view of this project it is hoped that entomologists and other scientists will be the eventual customers of this project.

The general design approach used by the mechanical group 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 [7]. 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 [7]. 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. Images of these two groups of insects can be seen in figure 1.

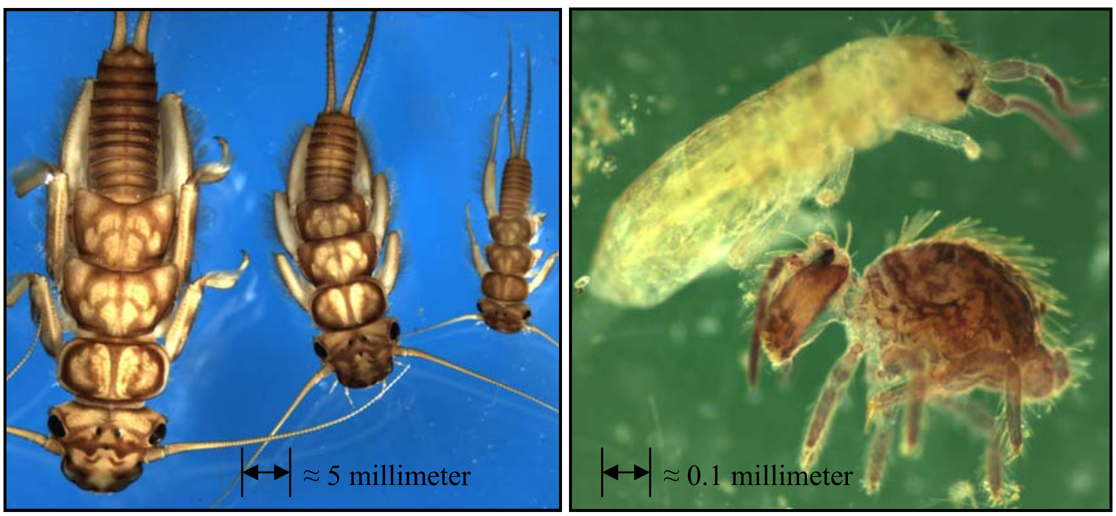

Figure 1 Stonefly larvae on the left and soil mesofauna on the right

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, a 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 an ethanol solution 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.

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 or green 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.

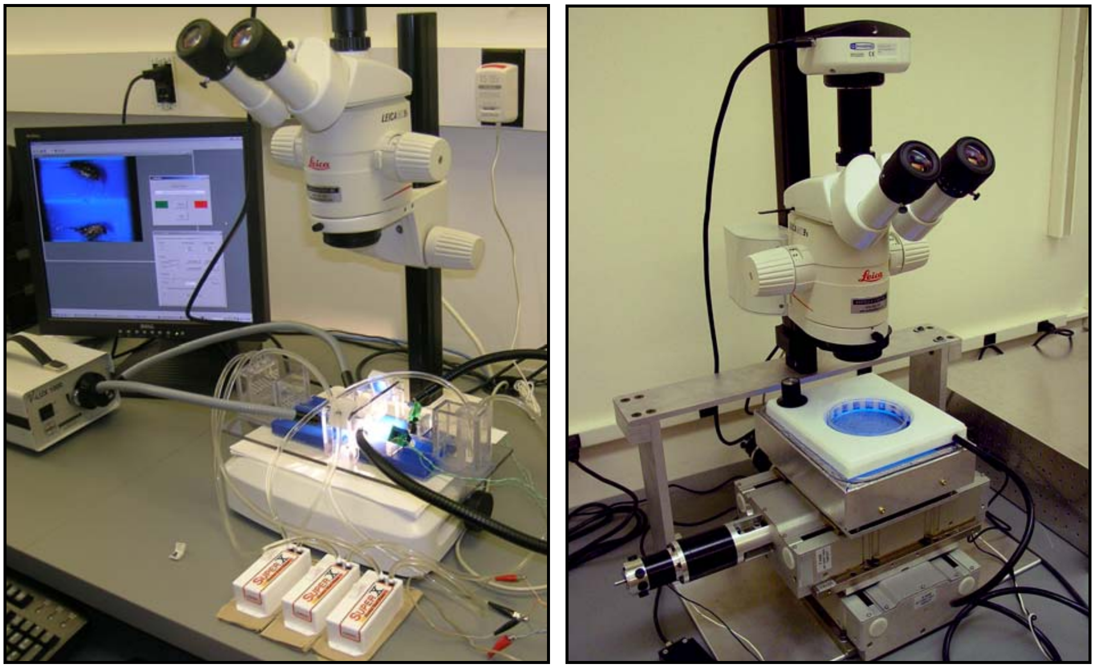

Figure 2 Stonefly larvae apparatus on the left and soil mesofauna apparatus on the right

The ultimate goal of the project is to have a new product that meets all of the customers' needs, but even at this stage, all the specifications are still not known [8]. 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 unsure exactly what they require for 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 three 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.

3 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 [9]. 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 [10]. The field of software design has investigated this problem to a much greater extent than that of mechanical design and 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 with positive and negative impacts [11]. This viewpoint is similar to that of mechanical design engineers and some management methodologies may be translated across disciplines. However, like with software design, project management does not specifically answer the question of how to follow a design process that is scope-creep-friendly.
Changing requirements has been a major problem in the growing field of software engineering. Software is often the most flexible and easily changed component of a system [12]. 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 with different mechanical interactions [13].
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 [12]. 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.
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. 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 [9].
Finally the idea of guessing future changes in software design can be applied to mechanical design. A robust design is one that can cope with alternative futures [14]. 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. Building flexibility into a system can be beneficial but is often expensive so it is important to determine where best to build flexibility into the system [14]. The target lifespan will determine how much flexibility the system should have to meet that target.
Research in mechanical engineering design 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 engineers in allocating time and resources [12]. Tracking changes can also give an engineer valuable experience in designing future products. 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 systems 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. A few approaches have been developed to assess the impact of a design change.
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 [13]. Reliable change propagation information is important for successful change management. 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 [13].
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 [4]. The DFV 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 [15]. 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 [15].

Freezing requirements is one way designers 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 [10]. A design freeze marks the end of a development stage where requirements become fixed before the design can continue [10]. 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 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 component 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 [10]. Many designers feel it is best to keep parts flexible where changes are anticipated. 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.

4 PRODUCT DEVELOPMENT STRATEGY FOR CHANGING REQUIREMENTS

In light of managing changes in design, based on experiences with the new product development in the Bug ID project and previous research on the topic, the mechanical group has identified and implemented the following six strategies described in the subsequent sections; 4.1–6. In sharing these experiences with others, it is hoped that the presented strategies can facilitate similar projects in a successful product design.

4.1 Establish and Foster Open Communication between Designers and Customers

Coupling active customer participation with rapid prototyping methods is the foundation for identifying unanticipated customer requirements in a timely fashion [4]. Designers face numerous obstacles that may hinder a product design. Errors in designs often are the result of mis-communication between domains, rather than within the domains where designers are experts [16]. Effective communication between the designers 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 [4]. To help facilitated 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 the 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. There may also be perceived barriers that emerge from the history and culture of the group and the individuals involved [4].

Fostering open communication requires strong social relationships built on trust between customers and developers. This approach was used extensively in the Bug ID project among the interdisciplinary partners, and it was particularly effective in both developing requirements and enhancing communication. 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 Explicitly Write Them Down

One important aspect of the design process and change management is the development of customer requirements [3]. Customer needs are expressed as written statements that are developed by interpreting the information gathered from customers [2]. 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 [4].

At times customers do not know what they want, and requirements may be underspecified. Customers 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, as stated in 4.1 [4]. Then 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 potential effects of change to the overall system [6]. 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 [6].

Finally, for iterative designs and product evolution, a complete list of requirements should be explicitly written for future evaluation. It is helpful to decompose requirements into component parts and structure the requirements so they are easy to follow and analyze. It is also important to identifying and define inter-requirement rules and relationships between different components [6]. 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.

A few obstacles to avoid when developing design requirements are over specification and omitting customers. 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 by an LCD screen component.

The design must also account for all the customers of the product [4]. Often in product development the initial end users are the designers, as they test the product for future improvements. Customers may also be other designers who are responsible for designing coupled components. Any way to facilitate this process will be beneficial to the product. There were issues in the Bug ID project with trying to match the software used to control the camera with control for the *x-y* stage that held the LCD screen. Initial stipulation for these components would have saved design 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.

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. Translate 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. Recognizing which requirements are likely to remain in future iterations will direct the direction of design. Some components will be more fundamental and should be given more attention.

It is important to analyze the design and determine which components satisfy which requirements. Requirements can then 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 these 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. Different components for stage translation, background color, and additional lighting could then be tested. The base platform was designed to be robust and flexible to accommodate future components. It has not been altered in several apparatus iterations.

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 [10]. 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 designers.

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 [14]. 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 [5].

In the Bug ID project, it was known early on that a blue background would be necessary for the images, however it was now known how bright or how blue the background would need to be. There was also talk about using a green 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. The general design approach in the Bug ID project (see Section 2) 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 needs quick turnover of designs and prototypes. Evaluation and frequent prototyping can help identify conflicts throughout the design process [16]. 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 definite 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, for instance, when the design for the stonefly larvae apparatus was started, 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 2, to alter the viewing angle and direction. All of this information will be useful for design of the next iteration.

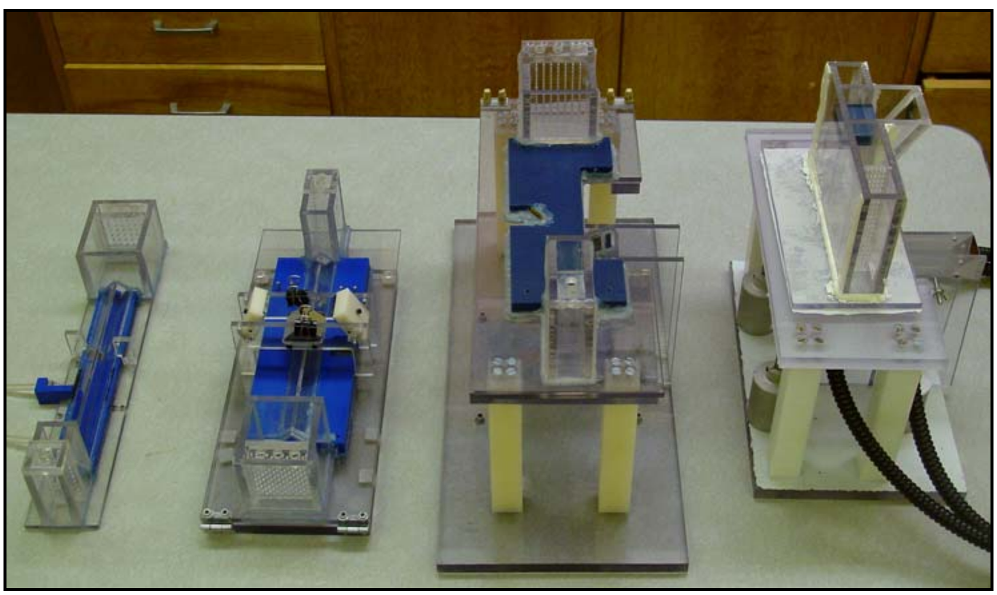

Figure 3 Several iterations of the stonefly larvae transport mechanism

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 [17]. 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. These 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. Designing the current product so that a component can absorb a large change in a specification before requiring redesign will decrease the possibility it will not be changed [15].

Specifications should be analyzed to determine if they are likely to change. Requirements that are likely to change should be translated into flexible components. 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 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.

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 [4]. An assessment will have to be made whether the benefits of flexibility outweigh the cost to functionality.

5 SUMMARY AND FUTURE WORK

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.

Through the development successes and failures of the Bug ID project, the mechanical design team gained valuable insight into the process of designing under changing requirements. Six product development strategies to cope with changing requirements and specifications were identified and applied to the project's evolving requirements design environment. These strategies were tested while developing working product prototypes for the project. This paper offers these six recommendations for designing with changing requirements to be applied to similar product development projects.

1. Establish and foster open communication between customers and designers. 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.
3. Analyze the list of requirements to identify which requirements are likely to change and which are stable.
4. Predict future customer needs and requirement 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.

Over the next few years, the above strategies will continually be applied to the development of mechanical design solutions for the Bug ID project. The immediate research plan is to use them to expedite the design cycle of the next generation of each apparatus. It is also a goal to establish a more systematic design methodology for designing under changing requirements, based on these strategies and the next generation designs for the project, with consideration given to change propagation throughout a system. The systematic methodology is expected to be applicable to many interdisciplinary product innovation scenarios and could provide a process for dealing with changes in design.

ACKNOWLEDGEMENT

The presented work is sponsored under the funding provided by the National Science Foundation (United States) Ref. 115-0326052.

REFERENCES

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

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

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

[4] 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.

[5] 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.

[6] 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

[7] 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).

[8] 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

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

[10] 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.

[11] Gray, C.F. and Larson, E.W. *Project management: The managerial process, Third Edition*., 2006. (McGraw-Hill).

[12] 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.

[13] 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.

[14] Land F. Adapting to Changing User Requirements. *Information and Management*, June 1982, 5(2), pp. 59-75

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

[16] 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

[17] 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.

Contact: R. K. Paasch
Oregon State University
Department of Mechanical Engineering
204 Rogers Hall
Corvallis, OR 97333
USA
Phone: 541-737-7019
Fax: 541-737-2600
paasch@engr.orst.edu
http://www.engr.orst.edu/~paasch