12-2013

Understanding the Role of Requirements in Engineering Design by Novices

Shraddha Joshi
Clemson University, shraddj@g.clemson.edu

UNDERSTANDING THE ROLE OF REQUIREMENTS IN ENGINEERING DESIGN BY NOVICES

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Mechanical Engineering

---

by
Shraddha Premjibhai Joshi
December 2013

Accepted by:
Dr. Joshua D. Summers, Committee Chair
Dr. John DesJardins
Dr. Georges M. Fadel
Dr. Gregory M. Mocko

ABSTRACT

Requirements play a critical role in the design process. The broader impact of this research is to develop a systematic understanding of the current use of requirements with an ultimate goal to develop guidelines and recommendations for more effective use of requirements throughout the design process. Thus, this research begins to answer the question about what is the role of requirements in design process and, specifically, its role in idea generation? The answer to this question is explored in three phases.

The first phase is to understand how requirements are currently taught to students. To that end, two surveys were conducted. First, a review of ten design textbooks was conducted as an initial surrogate for understanding what is formally taught. This was done to understand the use of requirements within the design tools mentioned in the textbooks. Supplementing this, interviews of faculty involved in teaching design courses was conducted with faculty from mechanical engineering, industrial engineering, bioengineering, and materials science and engineering. While the interviews suggest that the use of requirements is distributed throughout the design process, in agreement with common practice, the instruction provided students, based on the survey of textbooks, focuses on requirements tools found exclusively in the conceptual design phase. Thus, a significant gap is identified in terms of lack of sufficient tools explaining the use of requirements.

In order to understand the consequences of lack of tools and to develop a deeper understanding of how students are applying the requirements education imparted to them,

a case study analysis was conducted with senior mechanical engineering design students in a capstone course. Data was collected from four teams working in parallel on the same design project in form of requirements documents from initial weeks and the final report deliverable. The findings from this study reveal that there is lack of uniformity in how students elicit requirements in the initial weeks of the project. The completeness and specificity of requirements increase from the initial weeks to the final week, as expected, as the students develop a better understanding of the problem. However, in terms of addressing the requirements, more requirements with one adjunct or numerical value, and thus low specificity, were addressed. Further, it was found that the requirements documents of novice designers (students) change in multiple ways. Currently, the students do not have tools or methods in place that would allow them to systematically manage the changes in requirements document.

Finally, as a deeper dive into how requirements can impact a specific design activity, an empirical designer study was conducted to explore the impact of requirements elicitation in idea generation. The study was conducted, again, with senior mechanical engineering students at Clemson University. The findings from the experimental study suggest that the students elicit more non-functional requirements compared to functional requirements. However, the ratio of the number of non-functional to functional requirements decreases when considering only the requirements addressed during ideation. Further, comparing the requirements addressed in the solutions generated by the students, it is found that the group that was not primed with the task of eliciting requirements performed better in terms of addressing requirements when compared to other two groups.

Ultimately, the findings from these studies are used to make several recommendations that will allow the students to systematically use the requirements at various design stages and enhance their current use of requirements. This dissertation presents both broad and focused research evidence with respect to the role that requirements play in engineering design based on student experiences. This does not imply that professionals behave in a similar manner. However, as the understanding of requirements in the education of the students is further developed, this can have a significant, albeit indirect, impact of the practice in industry as the students graduate.

DEDICATION

This thesis is dedicated to:

The Almighty,

My parents Premji and Jyotsna, for bringing me in this world,

My soul mate Parikshit, for loving me and holding me through all the storms,

My oncologist Dr. Rajiv Malik, for bringing me back to life.

ACKNOWLEDGMENTS

I have no words to express my gratitude towards my advisor Dr. Joshua D. Summers. After I completed my master's degree, I knew I was not done learning from him and that motivated me to pursue my doctoral studies. I would not be what I am today without the guidance of Dr. Summers. He is the best teacher and advisor I know and I consider myself blessed to have him as my academic father. Be it formal advisor meetings in his office or more open discussions while having a cup of coffee or long distance Skype meetings, any chat with Dr. Summers is highly thought provoking and a great learning experience in itself. Dr. Summers gave me the freedom to pursue my research interest. He continued to believe in my abilities even when I highly doubted them and always encouraged me to perform my best. I highly admire his selfless dedication to his students. Be it 7 A.M in the morning or 10 P.M. in the night or a Sunday afternoon, Dr. Summers was always available to discuss and help me with my research. I am not sure if thank you would suffice for the gratitude that I feel towards him for moulding me to perform beyond my capabilities. Dr. Summers will always continue to inspire me by being an outstanding teacher, a great researcher and one of the best human beings. I hope one day I am able to inspire my students as much.

I would like to thank my dissertation committee members Dr. Gregory M. Mocko, Dr. Georges M. Fadel and Dr. John DesJardins for providing valuable guidance through the review of this dissertation. Their pursuit for excellence in all that they do is greatly inspiring.

I would like to thank my extended family at CEDAR lab for being with me though all the odds. I would like to thank Dr. Beshoy Morkos, Dr. Chiradeep Sen and Dr. Prabhu Shankar for being great friends, setting high standards of research to look up to and providing valuable criticism. The time we spent in lab discussing research, writing papers or simply sharing jokes will always be close to my heart. I would like to thank Emily Worinkeng, Liz Matthew, Rahul Renu, Keith Phelan and Mohammad Fazelpour for helping me at various stages of my research. Special thanks to Luke Berglind, Jesse Shultz and Aravind Shanthakumar for being the best "next chair neighbors", graduate school would not have been fun without you.

While pursuing graduate school is challenging for most students, I did not anticipate that I will be faced with cancer to add to those challenges. It was scary and overwhelming. There were days when I was too tired, awfully sick and just wanted to give up. In spite of all the odds, the love and support of people in mechanical engineering department at Clemson University helped me great deal to finish my dissertation successfully. I am forever indebted to the professors and staff of mechanical engineering department for believing in me and helping me in all possible ways. It was their belief in me that kept me going. Special thanks to Gwen Dockins, Tameka Boyce, Teri Garret, Janeen Putman and Kathryn Poole for being my "guardian angels" throughout my stay at Clemson University.

Finally, I want to thank you Clemson for being my home for five years and teaching me the most valuable lessons of life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE: MOTIVATION FOR RESEARCHING REQUIREMENTS IN IDEATION

In a systematic design process, the problem definition is followed by requirements elicitation [1]. Success or failure of design is determined based on whether the requirements are met or not, thus requirements play a critical role throughout the design process. Further, requirements greatly influence activities such as generation, testing and validation of design concepts [2, 3]. Although the importance of requirements is widely recognized, major failures result due to incomplete or inadequate requirements [4, 5]. Research also reveals that requirements that are ignored throughout the design process are the ones that remain unsatisfied [6].

Thus, the importance of requirements is highly recognized and researchers are investigating to improve various aspects of requirements. Further, two broad aspects are investigated: 1) requirements as a statement and 2) requirements as a document. Apart from these two broad aspects, researchers are also investigating the reasoning aspect of requirement which essentially looks at what to 'do' with requirements. Section 1.1 discusses the overview of requirements research.

1.1 Requirements Overview

This section provides the overview of requirements research. First, section 1.1.1 describes the various definitions of requirements as found in literature. Then, section 1.1.2 discusses the current research areas within requirements and finally section 1.1.3 describes the overview of requirements research within Clemson Engineering Design

Application and Research (CEDAR) lab. This ultimately leads to the motivation for this dissertation.

1.1.1 What are Requirements?

Requirements are defined as statements describing the goals that must be satisfied by the final design solution [1]. Essentially, the requirements include descriptions of the functions that a solution must perform and the characteristics or properties that a solution must possess [1, 7]. Thus, requirements represent the needs of a customer that must be fulfilled by the solution [8, 9]. Requirements are often classified as constraints and criteria [1]. While the constraints are the requirements that 'must' be fulfilled by the final solution, criteria are used for selecting a solution among various alternative design solutions [1, 10, 11, 12]. As another perspective, requirements are also classified as functional and non-functional. A functional requirement indicates what the system must 'do' and a non-functional requirement indicates the characteristics or properties that a system must possess [12, 13, 14].

A requirements document is established at the beginning of the design process through the process of eliciting requirements [5, 15]. This document is maintained and updated throughout the design process [1]. Requirements not only serve as input for generating conceptual ideas for a given design problem at the beginning of the design process but are also used for validation and testing the concepts at the end of the design process [7]. Thus, requirements play a critical role throughout the design process.

Next, Section 1.1.2 provides an overview of the various aspects of requirements that are currently researched.

1.1.2 Summary of current requirements research

As previously mentioned, the requirements research can be broadly classified into two aspects 1) studying requirements as 'statements' and 2) studying requirements as 'documents'.

It is evident from literature that requirements play a critical role from the beginning to the end of design process. Successful completion of the project results from fulfilling the requirements established at the beginning of the project [16]. Requirements are often expressed in natural language sentences [17]. It is therefore necessary to ensure that the requirements are correctly written to mitigate ambiguities associated with incorrect or incomplete requirements [17]. This leads to motivating the research associated with a requirement as a 'statement'.

Moreover, writing 'good' requirements is a major area of investigation [17, 18]. A 'good' requirement is a statement that is necessary, verifiable, and attainable [18]. Thus, within a requirement document, if a statement is not necessary, cannot be tested, or met within the limitations of time and budget, then it is not a 'good' requirement.

Further, it is necessary that a requirement be of 'good quality'. While the quality of requirements is little explored within mechanical engineering domain, various quality attributes of requirements are established in literature pertaining to software engineering. Thus, a 'good quality' requirement is the one that is complete, consistent, correct, modifiable, ranked, testable, traceable, unambiguous, validatable, and verifiable [19, 20].

Writing a 'good' requirement is explored from linguistic perspective by mapping the components of a natural language sentence to the components of a requirements sentence. “Completeness” of requirements is thus measured in terms of missing components within a requirement statement [17]. Further, linguistic semantics are also applied to improve the computational understanding of requirements statements [21].

Establishing the classification of a requirement statement into functional and non-functional is another area of investigating requirement as a statement. It is necessary to establish this classification as it relates to the level of detail within a requirement statement [12]. Functional requirements are the requirements that describe what a system must ‘do’ [13, 14]. On the other hand, the non-functional requirements describe the characteristics that a system must possess [13, 14]. Further, the non-functional requirements are typically derived from functional requirements [12]. Again, within mechanical engineering domain, efforts have been made to establish this classification based on the linguistic analysis of the requirement [17]. Thus, linguistically, a requirement sentence having the main verb as transitive or intransitive is defined as a functional requirement and the requirement sentence having the main verb as a linking verb is non-functional requirement [17].

The second important aspect of requirements research is found in investigating issues associated with requirements as a document. As previously mentioned, the requirements document is generated at the start of the design project when requirements are elicited. The document evolves throughout the design process as new information is added to it [22]. Thus, it is important to have tools and methods that would help to manage the requirement document, specifically addressing the changes. If the changes within the requirements document are not properly managed, they could pose a difficult and costly problem [23, 24]. Further, to ensure that valuable information within a requirement document is not lost, it is necessary to have tools and methods that allow tracing and managing the changes within requirement documents. Within mechanical engineering domain, efforts have been made to develop tools that would allow predicting requirements change [16].

From the literature, it is evident that requirements play a critical role within design process and this necessitates researching various aspects of requirements to improve the quality of requirements and manage requirement documents. This will then aid in fulfilling the requirements leading to successful project outcomes. Next Section 1.1.3 provides an overview of requirements research within Clemson Engineering Design Application and Research (CEDAR) lab.

1.1.3 Requirements Research in CEDAR

Researchers within CEDAR lab at Clemson University have made significant contributions to the various aspects of requirements research. Figure 1-1 illustrates the overview of requirements research conducted within CEDAR lab at Clemson. Different aspects of requirements researched within CEDAR include research pertaining to requirement statement, requirements document and requirements reasoning.

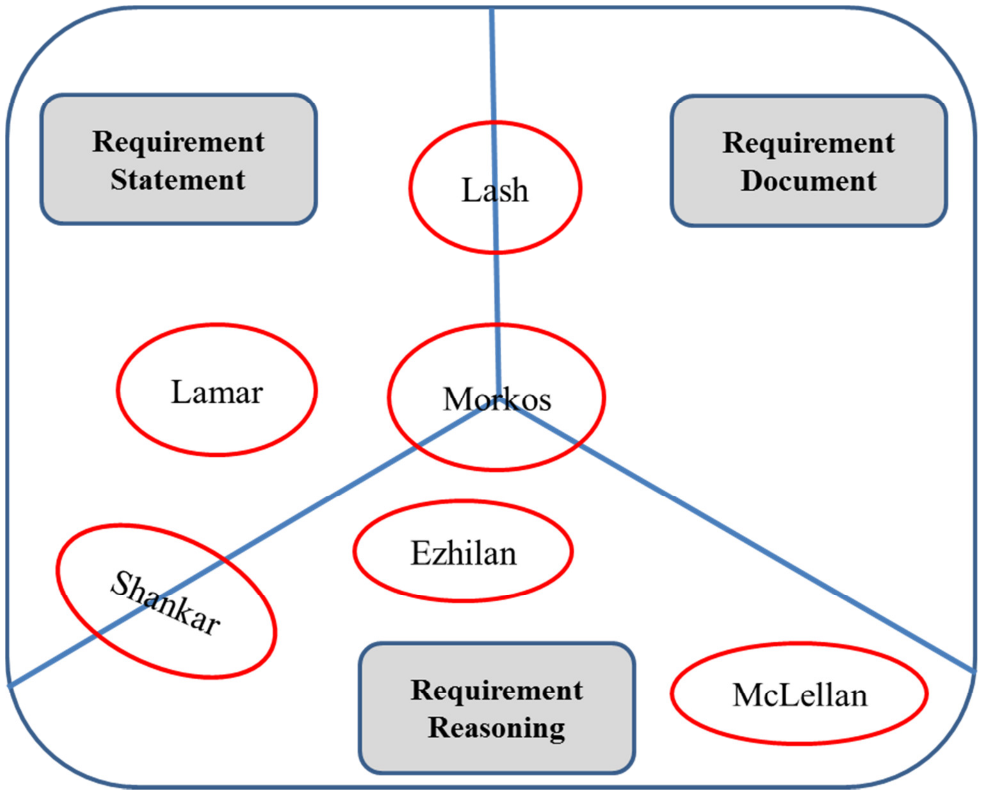


Figure 1-1 Requirements research at CEDAR

Efforts have been made to measure the ‘quality’ of requirement through linguistic analysis of requirements [17]. Linguistic analysis is also used for improving computational understanding of requirements by identifying and forming semantic relationships between requirement statements [21]. Further, researchers within CEDAR have developed tools for predicting requirements change [16] and reducing the changes due to change propagation effects [25]. Finally, requirements analysis is used as a method for mass reduction in solutions [26, 27]

Thus, section 1.1 provides an overview of various areas of investigation within requirements. While the researchers are investigating tools and methods to improve the quality of requirements and managing requirements documents, noticeable gaps exist in current literature with respect to understanding how designers use requirements after elicitation. Figure 1-2 illustrates the motivation for researching requirements in conceptual design.

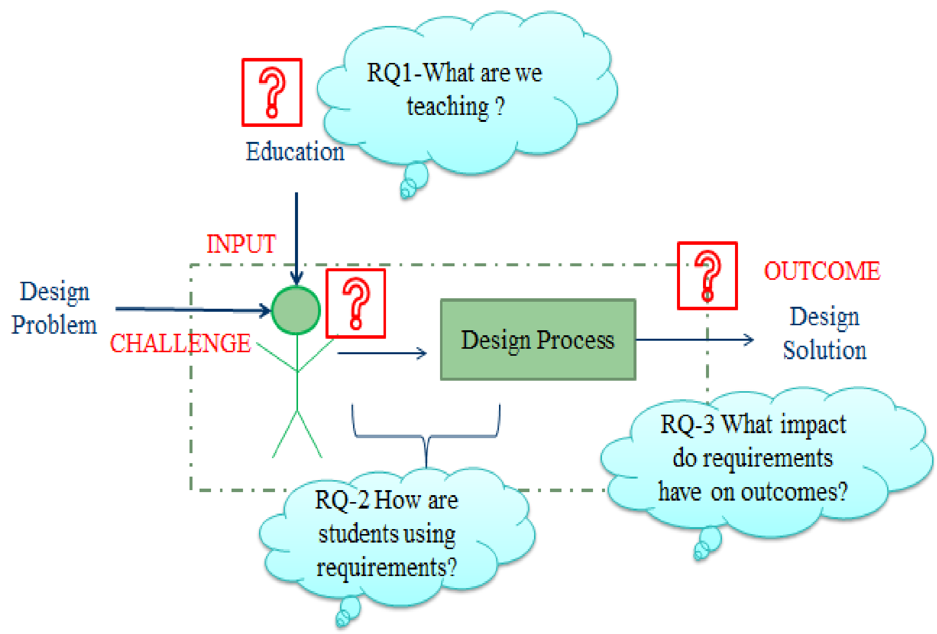


Figure 1-2 Motivation for researching requirements in conceptual design

As shown in Figure 1-2, for a novice designer, the design education imparted in the classroom serves as knowledge input. This knowledge input gained through classroom learning and design textbooks is critical because novice designers do not have much real world experience and thus rely heavily on what they learn in the classroom. Therefore, it becomes necessary to investigate what is currently taught to students in terms of various aspects of requirements.

When faced with a challenge in form of a design problem, the novice designers rely on the design process taught in the classroom to develop a potential solution. Applying the various stages of design process, the novice designers develop solutions for a given problem. In order to ensure that the students are correctly applying the process, it then becomes necessary to investigate how novice designers are currently using requirements. Finally, to facilitate the development of better solutions it becomes necessary to investigate how students can use requirements in conceptual design.

It is important to understand the role that requirements play at each stage in the design process if educators are to help train effective designers. This knowledge is critical for novice designers, especially graduating engineering students, as they do not possess past experiences on which to rely. A preliminary study was conducted to explore and understand how requirements are currently taught to novice designers, with specific emphasis on their use within the design process, as evidenced through typical undergraduate engineering design textbooks in mechanical engineering in the USA. The findings suggest that, though most design tools in the textbooks mention the use of requirements either explicitly or implicitly, they lack the rigor in describing the specific details such as type and number of requirements and requirements selection strategy that should be used.

Further, the importance of eliciting, documenting and validating requirements has been realized and researched [28, 29] but little research exists in the area of understanding the specific roles that requirements play at various stages after they are elicited. To that end, this research begins to answer the overall question – <u>what is the</u>

role of requirements in conceptual design, specifically idea generation? In order to investigate the role of requirements in conceptual design, three questions are investigated through this research:

- RQ1 What are we teaching?
- RQ2 How are students using requirements?
- RQ3 What impact do requirements have on outcomes?

The answers to these questions will begin to address overall objective of this research - to provide systematic guidelines for design instructors and practicing designers for using requirements in conceptual design.

1.2 Dissertation Scope Resolution

The scope of this research is limited to generative design problems. The stages of interest include primary task clarification and conceptual design, though embodiment and detailed design are found in the case study. Further, the requirements are mostly centered on constraints. However, they may also be "soft" constraints that are generally used for comparison of different alternative solutions as if they were criteria. The scope is limited to senior students/novices who will be practicing engineering within six months of study. This study is not explicitly focused on practicing engineers.

1.3 Dissertation Roadmap

This section discusses the overview of the dissertation. This dissertation is organized in six chapters. Chapter Two discusses the research questions, justification for investigating each research questions and summary of findings from each research question. Chapter Three discusses the details of RQ1-what are we teaching? It describes the protocols for survey of design textbooks and survey of faculty. The findings from both the surveys are then summarized to identify critical gaps within requirements education. Chapter Four discusses the details of RQ2-how are students using requirements? To investigate RQ2, a case study was conducted with senior design students at Clemson University. Chapter Four describes the details of the case study such as design problem, data collection method, and data analysis protocols. The findings of case study to understand how students are currently using requirements are then explained followed by the limitations of case study. Chapter Five discusses the details of RQ3-What role can requirements play in idea generation? To that end, a designer study was conducted with senior design students at Clemson University. Chapter Five describes the specific details of designer study such as design problem, description of participants, execution procedure, data collection method and protocols for analyzing the collected data. The summary of findings for RQ3 is then discussed in this chapter. Chapter Six discusses the conclusions of this dissertation along with the intellectual merit and broader impact of this research. Finally the future directions of this research are discussed.

CHAPTER TWO: SUMMARY OF RESEARCH QUESTIONS AND FINDINGS

The goal of this proposed research is to understand the current state of use of requirements in conceptual design, specifically as taught and practiced in mechanical engineering design education. After understanding the current state and identifying the gaps in teaching and application, the ultimate goal is to be able to provide systematic guidelines to instructors and designers for using requirements within idea generation. Figure 2-1 illustrates the overview of proposed research.

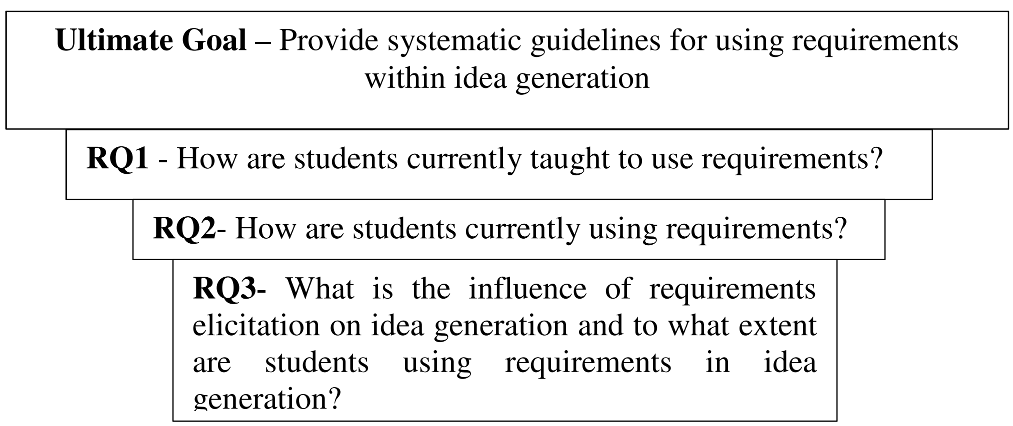


Figure 2-1 Overview of Research

The overall research objective to understand the current state of requirements use and providing requirements guidelines for idea generation was realized by answering three broad research questions. Four specific tasks are defined to answer these research questions. Table 2-1 illustrates the research questions and the related tasks.

Table 2-1 Research Questions and Tasks

Research Questions	Research Sub-questions	Tasks
RQ1-How are students currently taught to use requirements?	RQ1.1 How do the design-tools taught at different phases of design process in engineering design courses use requirements?	Task-1A: Survey of design textbooks Task-1B: Survey/Interview of faculty
	RQ1.2 How is the use of requirements within design tools explained and described in the textbook?	
	RQ1.3 What 'requirements related activities' are performed by the students as evident from surveying faculty?	
RQ2-How are the students currently using requirements?	RQ2.1 What is the influence of change in number of requirements elicited by individual teams and those provided by the sponsor on level of detail of final solution?	Task-2 Case Study with ME-402
	RQ2.2 What is the completeness and specificity of requirements in initial week and final week?	
	RQ2.3 What is the specificity of requirements met in the final solution?	
	RQ2.4 What is the evolution of requirements as evident from requirements tracing?	
RQ3 – What impact do requirements have on outcome?	RQ3.1 When assigned the task of eliciting requirements, what types of requirements are elicited by novice designers?	Task-3 Designer Study
	RQ3.2 Does the involvement of designer in the task of elicitation impact the number of requirements addressed by them? How does this compare to providing them with a list of requirements?	
	RQ3.4 Does having a list of requirements cause fixation when addressing more requirements?	

Three major activities were conducted to address this overall goal. First, a review was conducted to understand how requirements education is currently imparted to students. This included surveying the design tools discussed in mechanical engineering design textbooks and surveying faculty actively involved in teaching design in the undergraduate curriculum. This review will help to identify critical gaps in current requirements education.

The next step was to conduct a case study on a senior design project at Clemson University to investigate the shared understanding of requirements across design teams working on the same design project. This case study will also serve as a validation of whether or not there is uniformity in application of the requirements education imparted to students.

Finally, a designer study was conducted to understand the influence of requirements elicitation activity on the idea generation process. The goal of this study was to identify the difference in the extent of requirements addressed between the designers who elicit the requirements and those who are provided with requirements list beforehand. The detailed justifications for the research questions, the related tasks to answer these questions and the summary of the findings for each are discussed in Sections 2.1, 2.2 and 2.3.

2.1 RQ1: What is Currently Taught?

RQ1 aims at exploring what is currently taught to students in terms of requirements education. The justification for exploring this question is discussed in section 2.1.1. Section 2.1.2 describes the tasks for answering RQ1 and finally section 2.1.3 discusses the summary of findings for RQ1.

2.1.1 RQ1: Justification

Requirements education is meant to prepare students to use requirements in design practice. An adequate understanding of requirements and various activities associated with requirements such as elicitation, documentation, and use throughout the design process is necessary to prepare the students for real world situations. Further, the ability to be able to design products that meet customer needs has also been identified by ABET as an important criterion for graduating engineers [30]. Capstone design courses play a major role in providing real world experience to graduating seniors [31]. These courses provide the students with opportunities to practice the classroom learning by executing a real world problem. The type of instruction provided to the students during the capstone course varies from few lectures at the beginning and project execution for the remainder of the semester to intermediate lectures throughout the semester [32]. Of the various design related tools and methods taught to the students during the capstone design course, teaching them aspects related to requirements play a major role [29, 33] as understanding the problem at hand is the first step to successful design [34, 35]. Thus, it becomes necessary to investigate whether or not the goal of teaching requirements is adequately met through current requirements education. To that end, RQ1 aims at understanding how requirements are currently taught to the students. Further, use of design textbooks and classroom learning are two important knowledge inputs for students. Thus, in order to investigate how requirements are currently taught to the students both these inputs are explored. The details of the tasks for answering RQ1 are further explored in section 2.1.2.

Further, it may be noted that the surveys of textbooks and faculty were conducted with the main purpose of identifying the gaps in teaching of requirements. Thus, this is not an education centric study with focus on details of instruction methods. Rather, the goal of this study is to obtain general understanding of what types of material are taught with respect to requirements.

2.1.2 RQ1: Tasks

In order to understand how requirements are currently taught to students, two tasks were conducted: Task-1A – Survey of design textbooks and Task-1B – Survey/Interview of faculty specifically instructing design related courses. This review provides both the motivation and the foundation for understanding requirements as taught in mechanical engineering curricula. The first task provides a coarse overview of what is likely to be taught in the design preparation courses through the lenses of the textbooks that have been explicitly developed for this purpose. The primary focus was on surveying the design textbooks, specifically focusing on how different design tools used within the textbook incorporate requirements. It may be noted that textbooks are used initially as surrogates for classroom teaching.

The second task delves deeper into exactly how the topic of requirements in design is addressed in the classroom. This was done through a survey of faculty teaching design related courses. The survey was designed to gather information about the requirements related activities and tools that students are anticipated to use throughout the design process.

The findings from both these surveys are used to identify critical gaps within requirements education. While the details of the execution of the tasks and findings are discussed in Chapter Three, section 2.1.3 provides a summary of findings for RQ1.

2.1.3 Summary of findings for RQ1

RQ1 aims at investigating what is currently taught to students in terms of requirements education. A survey of design textbooks and survey of faculty involved in teaching design was conducted as surrogate for classroom teaching.

First, a survey of ten design textbooks was conducted to investigate the tools using requirements. Then, how each of these textbooks describes the use of requirements within various design tools was investigated. The findings from this survey indicate that while most design textbooks identify the importance of requirements, they lack the discussion on what to do with requirements after they are elicited. Further, there are ambiguities related to how to use requirements within design tools. Most design textbooks fail to mention the details such as number of requirements, type of requirements, strategies for selecting requirements for use in particular design tool. Thus, it is left to the discretion of the user to make these decisions.

The findings from the faculty interviews and surveys reveal that requirements related activities are performed throughout the execution of the design project. From the survey of the design textbooks, it is evident that maximum numbers of tools mentioning the use of requirements are found in the conceptual design phase. There are fewer tools mentioning the use of requirements in other phases of the design process. Thus there is a significant gap in terms of lack of sufficient tools that would allow the students to use the elicited requirements throughout the design process. Next, section 2.2 discusses the justification, tasks and findings for RQ2.

2.2 RQ2: How are students using requirements?

After investigating what is taught to students in terms of requirements education, the goal of RQ2 is to explore how students are apply the requirements education imparted to them and thus identify the gaps within the application of requirements. First, section 2.1.1 discusses the justification for exploring RQ2. Then, section 2.1.2 describes the task conducted to answer RQ 2 and finally section 2.1.3 provides a summary of findings for RQ2.

2.2.1 RQ2: Justification

After understanding how requirements are currently taught to students, it becomes necessary to investigate how students are applying the requirements education to real world design problems. For novice engineers, who lack the necessary experience, classroom education plays a major role of providing with necessary information and training to execute real world design projects. However, what is taught in classroom may not be conveyed in the manner that is desired and, thus, it is possible that graduating engineers do not have the necessary requirements education. This can lead to serious problems when novice engineers are tasked to work on a real world design problem. This leads to the necessity of investigating the application of requirements education by novice engineers and thus the second research question – *how are students currently using requirements?*

Further, four important aspects are considered to understand how students are currently using requirements. First, the ‘delta’, or change, in requirements between student teams and sponsor is mapped to the level of detail of final solution to investigate if missing the sponsor requirements or eliciting more than given requirements affects the level of detail in the final report. Second, the completeness and specificity of the elicited requirements is explored to identify if the students have the required tools and methods for eliciting requirements. Third, the specificity of requirements met in the final solution is investigated. This will help to identify the level of detail required for the requirements that are successfully met in the final solution. Finally, the evolution of requirements is investigated to explore the type of changes in the requirements document of novice designers.

The findings for RQ2 are then compared with those from RQ1 to identify further gaps from a combined point of view of imparting and application of requirements education. Next, section 2.2.2 provides an overview of the task completed to answer RQ2.

2.2.2 RQ2: Tasks

In order to answer RQ2, a case study was conducted with senior design student teams at Clemson University. Case study is used as research method to investigate a phenomenon in real-life context, specifically to answer ‘how’ and ‘why’ type of research questions [36, 37, 38]. Further, the phenomena under investigation could either be contemporary or historical. Some examples of use of case study as a research method in design can be found in [3, 39, 25, 40]. As previously mentioned, RQ2 aims at investigating how students are currently using requirements. Further, the goal is also to compare the requirements elicitation across design teams working on same project and thus the study of a senior design project is appropriate to answer this question as there are multiple teams working on the same project [41, 42, 43, 44, 45].

Each team is given an initial problem and a preliminary set of requirements from the sponsor in the first week of the semester. After that, each team works individually to elicit new requirements and develop solution to address given design problem. Thus, a case study with these design teams will help to compare –the requirements elicited by each team working on the same project. This comparison will then help to understand whether or not student teams have similar understanding of the design problem and requirements elicitation. A weekly requirements update sheet was created to collect the weekly requirements data from the teams. In addition to the weekly requirements update sheet, the final design reports generated by the students as a part of project deliverable will be used for this case study. Further details of this case study (Task-2) are provided in Chapter Four. Section 2.2.3 provides a summary of findings from RQ2.

2.2.3 Summary of findings for RQ2

RQ2 aims at investigating how students are currently using requirements. In order answer RQ2, a case study was conducted with senior design students at Clemson University through the capstone project in their final semester.

The findings from mapping the ‘delta’ in requirements to the level of detail of final solution reveals that a team with consistent delta has a high level of detail in the final solution, while the teams have either positive or negative delta results to a medium level of detail in the final solution. A positive delta in requirements essentially means that the students have elicited more requirements than those provided by the sponsor. While this indicates a better understanding of the problem, if the requirements are elicited late in the design process, then the students may not have sufficient time to address them, thus resulting to a poor detail in the final solution.

Next, the completeness and specificity of elicited requirements were explored. While completeness is measured by considering the linguistic components of requirements, specificity is measured as the level of detail in the requirements in terms of number of adjuncts and numerical values. The findings reveal that the teams have low completeness in initial week indicated by greater number of requirements missing either subject or modal. However, in the final week, the number of incomplete requirements decreases resulting to an increase in completeness of requirements. The same is true for specificity; the teams have more requirements with zero or one adjunct or numerical values in initial weeks compared to final week. As no specific or formal feedback is provided to students in terms of completeness or specificity, the increase could result from general feedback or increase in the understanding of the problem at hand.

Investigating the specificity of requirements met, it is found that the requirements with one adjuncts or numerical values will have a higher likelihood of being met in the final solution. Requirements that are too abstract (as indicated by either zero adjunct or zero numerical value) or too specific (as indicated by more than one adjunct or numerical value) pose difficult challenge and this will result to the failure in meeting them in the final solutions.

Finally, the requirements tracing study reveals that the requirement document of novice designer's change and the change occurs in multiple ways. Currently, the students do not seem to have appropriate tools to trace and manage these changes. This could lead to loss of important information pertaining to requirements and thus jeopardize the successful completion of the project. Thus, it is recommended that students should be taught appropriate tools to manage the project requirements and trace the changes. Perhaps, the weekly requirements update sheet can serve as a starting point to that end. Next, section 2.3 discusses the justification, tasks and findings for RQ3.

2.3 RQ3: What impact do requirements have on outcomes?

The goal of RQ3 is to investigate what roles requirements can play in idea generation. Section 2.3.1 provides a justification for investigating RQ3. The details of the task are provided in section 2.3.2 and section 2.3.3 discusses summary of the findings from RQ3.

2.3.1 RQ3: Justification

After understanding the requirements elicitation and usage in teams, the final goal is to understand requirements elicitation and use by individual designers. This leads to the third research question - *What is the influence of requirements elicitation on idea generation and to what extent are the students using requirements in idea generation?*

To that end, three aspects will be investigated through a user study 1) what types of requirements are elicited by individual designers when assigned with the task of requirements elicitation for a given design problem, 2) difference between individual designers in terms of addressing the requirements when provided with a list of requirements and when asked to elicit requirements, and 3) whether or not providing a list of requirements causes fixation while addressing requirements. This investigation is necessary as the findings will indicate whether or not designer's involvement in the process of requirements elicitation leads to a greater number of requirements addressed. If the findings from this study suggest that designers' involvement in the process of eliciting requirements has a positive influence on addressing more requirements in the concepts, then this knowledge can be used to develop new guidelines for teaching the use of requirements in idea generation. Additionally, new tools can be developed that will help designers to be able to use requirements as input and generate concepts as output. This will specially benefit novice designers who do not have sufficient experience in the field. Next, section 2.3.2 describes the designer study conducted to answer RQ3.

2.3.2 RQ3: Tasks

In order to understand the use of requirements in idea generation, a designer study was conducted with senior design students. The students were divided in three groups. While the students in all three groups were given the same design problem, one group was given ten requirements [10(no) group], one group was given five requirements [5(yes) group] and one group was not given any requirements [0(yes) group]. It may be noted that though the students were divided into groups based on experimental condition; they performed the task individually. The students of all three groups were asked to develop solutions to the given problem. In addition, the students given five and zero requirements were also asked to elicit more requirements before generating the solutions.

The data collection from the study entailed the requirements elicited by two groups and solutions generated by all three groups. The requirements elicited by the students were analyzed to investigate the type of requirements elicited and addressed.by them. Design solutions generated were studied to evaluate the number of requirements addressed by the students. While the details of this task are discussed in Chapter Five, section 2.3.3 provides summary of findings from RQ3.

2.3.3 Summary of findings for RQ3

RQ3 aimed at investigating the role of requirements in idea generation. To that end, a designer study was conducted with senior design students. Requirements elicited by the students and the sketches generated by the students were collected from this study and analyzed to answer RQ3.

First, the requirements elicited by the students were analyzed to investigate the type of elicited requirements. To that end, the requirements elicited by the students were classified into functional and non-functional. The findings from this study reveal that more non-functional requirements were elicited by the students compared to functional requirements. This was true for both the groups tasked with eliciting the requirements [5(yes) and 0(yes)]. While more non-functional requirements were elicited compared to functional requirements, the ratio of the average number of non-functional to the average number of functional requirements addressed decreases indicating that students perform comparably while addressing the two types of requirements.

Next, a comparison of the number of requirements addressed in the solutions between the three groups was made. First, the groups were compared for the number of given requirements addressed. These include the requirements embedded in the problem statement and the requirements in the list given to the students of two groups. Then, a comparison of number of unique requirements addressed was made. The unique requirements were extracted by considering the union of given and elicited requirements. Further, two types of comparisons were made, first considering each sketch as a data point and second considering each student as a data point.

The findings from comparing the number of given requirements addressed reveal that there is a significant difference in the number of given requirements addressed between the three groups when considering each solution as a data point. However, the difference is not statistically significant while considering each student as a data point. Further, on an average, maximum number of given requirements were addressed by the students of 10(no) group but this could result from the fact that the students of these groups were familiar with the given requirements. 5(yes) group addressed minimum number of given requirements on average.

Comparing the groups for the number of unique requirements addressed, it was found that there is a significant difference in the number of unique requirements addressed between the three groups when considering each solution as a data point. Again, the difference is not statistically significant while considering each student as a data point. Similar to the given requirements, on average, maximum number of requirements were addressed by the students of 10(no) group, while 5(yes) group addressed the minimum number of requirements on an average.

Both these studies reveal that on an average the students who do not elicit any requirements performed better in terms of addressing the requirements in the solution. While the students of both the groups tasked with eliciting more requirements perform poorly in terms of addressing the requirements in the design solutions.

This could stem from the fact that the students that were given the problem and list of requirements had a better understanding of problem at hand. The students tasked with eliciting requirements spent ten minutes further exploring the problem through requirements elicitation. They were limited to ten minutes for eliciting the requirements and at the end of this time, they may not have completed the exploration of the problem at hand resulting to poor understanding of the problem. This is then reflected, possibly, in fewer requirements fulfilled in the design solutions compared to the other groups.

Next, the fixation while addressing requirements was investigated. In order to do this comparison, the average ratios of number of requirements addressed to the number of requirements given were compared between the three groups. The findings suggest that 0(yes) group had maximum ratio of number of requirements addressed to the number of requirements given. Thus, they perform better while addressing the given requirements which for 0(yes) group are the requirements embedded within the problem statement. However, comparing these findings to the average number of unique requirements, it is found that on average, 10(no) group has more unique requirements addressed compared to 0(yes) and 5(yes) groups and this difference is significant. This indicates that the students of 0(yes) group are fixated on addressing given requirements.

CHAPTER THREE: REQUIREMENTS-WHAT ARE WE TEACHING?

3.1 Study Objective and Overview

The goal of conducting a critical review of design textbooks and survey of faculty is to understand the current state of requirements education, specifically in mechanical engineering undergraduate curriculum.

3.2 Requirements in Engineering Education

Requirements play a critical role within any design process as the activity of identifying and maintaining a system's requirements influences the success of a project [39]. Requirements are used at different stages within the design process after they are elicited as they support many subsequent activities [46]. For instance, requirements are needed for activities such as idea generation, concept evaluation, and concept selection. If asked, most undergraduate students will not recognize the importance of requirements throughout the design process; nor will they recognize the misuse of requirements as a major cause of project failure [22, 47]. Requirements, and how they are used, can have a significant effect on the success of a project, and the costs involved. However, unlike expert designers, novices, such as recent engineering graduates, do not possess past experience to judge the use of requirements at various stages of design. As a result, students rely heavily on their requirements education. Specifically, this chapter will focus on requirements education through design courses in mechanical engineering.

The process of capturing, analysing and tracking requirements throughout design, is of great significance [48], yet, there is minimal literature on teaching requirements

[49]. Students are typically taught different requirement activities over the duration of a project and are presented with a host of methods and techniques which they can use [33]. These methods and techniques can be taught directly by the instructor or through supplemental resources such as a textbook. Engineering design courses cover multiple design topics such as different approaches, methods, techniques, and tools while preparing students to complete a project. Unfortunately, the activities relating to requirements, though central to any project's success, often receive minimal attention [50]. Though most educators and professionals recognize the importance of requirements, it is difficult to prove the value of requirements to students [48].

Thus, it becomes necessary to explore following research questions:

- RQ 1.1 - How do the design-tools taught at different phases of design process in engineering design courses use requirements?
- RQ 1.2 - How are the use of requirements within design tools explained and described in textbooks?

In order to answer these questions, design texts, which incorporate sections for teaching its readers how to use requirements, are investigated. A survey of the engineering design textbooks is used as a surrogate for understanding the current best practices in teaching requirements. Ten design textbooks [1, 9, 51, 52, 53, 54, 55, 56, 8, 10] used in the undergraduate curriculum in mechanical engineering were identified and used for this survey. It is assumed initially that these books are used in the classroom and, therefore, requirement related education can be extracted from the texts.

3.3 Design of the protocol

A protocol is developed for each research question. The protocols answer questions pertaining to the design tools and the use of requirements within.

3.3.1 Understanding use of requirements in design process

The first research question aims at identifying the design tools which use requirements as taught to the students in the various design phases. A protocol is developed to classify where various design tools are used within the design process and identify those that make use of requirements. A four design phase model of the design process is used here [1]: (1) planning and clarifying the task, (2) conceptual design, (3) embodiment design, and (4) detailed design. The tools discussed in each textbook are classified based on the design phases, detailed in Table 3-1, in which the tools are used.

Table 3-1:Activities in Pahl and Beitz Design Process

Design Phase	Activities
Planning and clarifying the task	Forming design teams Generating product development plan Understanding design problem Developing customer requirements Assessing competition Generating engineering requirements Establishing engineering targets
Conceptual Design	Establishing function structures Searching for working principles and working structures Developing concept variants Evaluating the concepts against technical and economic criteria
Embodiment Design	Preliminary form design (includes prototypes) Performing design analysis Material and process selection Selecting best preliminary layout Refining and improving layouts Preparing preliminary parts list and production and assembly documents
Detail Design	Elaborate detail drawings and parts lists Complete production, assembly, transport and operating instructions Product documentation

The developed protocol is illustrated in Table 3-2. The first column indicates the four design phases. The second column in Table 3-2 notes the design tools used in each of the phases listed in column 1. Each textbook uses slightly different definitions of their discussed design phases. Thus, each design tool in the appropriate design phase with respect to defined four phase design process, rather than the phases that the original textbooks may have discussed. The third column indicates the use of requirements, which is further classified as explicit, implicit, and not mentioned. This classification further is explained below:

- *Explicit* refers to when design tool explicitly mentions the use requirements. For instance, the description of for morphological chart in [54] mentions "usually expressed in…product requirements or functions," and therefore it explicitly mentions the use of requirements.
- *Implicit* refers to when design tools do not explicitly mention the use of requirement, but it can be interpreted from the description of the tool. For example the description of brainstorming in [1] mentions "before actual brainstorming session, the leader must outline the problem…." The description does not directly state using requirements for idea generation. However, a problem statement is essentially a high level requirement and thus it can be interpreted from the description that requirements are used to generate ideas in brainstorming. Therefore, the use of requirement in the tool is implicit.
- *Not mentioned* refers to when the use of requirements is neither explicitly mentioned nor can it be interpreted from the description of the design tool. For

instance, fault tree analysis is a failure analysis tool which does not explicitly or implicitly mention using requirements to aid the analysis of failure.

Table 3-2: Use of requirements in design process

Phase in the design process	**Design Tools**	**Use of requirements** Explicit/Implicit/Not mentioned
Planning and clarifying task		
Conceptual Design		
Embodiment Design		
Detailed Design		

After developing the protocol, the design textbooks were read and thoroughly examined by at least two different examiners to ensure objective agreement.

3.3.2 Protocol for how requirements are explained and described

The second research question aims to understand how students are taught the use of requirements in the design tools which use requirements. The protocol illustrated in Table 3-3 is used to answer this research question. The descriptions of the design tools that explicitly or implicitly mentioned requirements were examined in order to populate the data. The first column in the protocol indicates the design tool under consideration. The remaining columns provide three different aspects that identify whether design tools, through its description in the design text book, address the use of requirements including: 1) types of requirement, 2) number of requirements and 3) requirements selection strategy. Each of these is further explained in detail.

Table 3-3 Aspects of teaching requirements

Design tool	Types of requirements used	Number of requirements	Requirements selection strategy
	Functional/ Nonfunctional/ Not mentioned		Systematic/ Random

- *Types of requirements*: Whether or not the description of the tool indicates the types of requirement to be used as an input is captured in column 2 of the protocol. For the purpose of this study, the types of requirements under consideration are functional and non-functional. Readers may consult [14] for details on the classification of functional and non-functional requirements. For example, the description of brainstorming in [9] mentions "A brainstorming session should be focussed on one specific **function**…" thereby indicates the use of functional requirement. Thus the entry for the column would be 'functional'. Specifically, function requirements are those requirements which describe the system's behaviour and performance while other requirements are considered non-functional [57, 58]. If the type of requirement to be used as input is not mentioned then the entry in the column would be 'not mentioned'.
- *Number of requirements*: The third column indicates whether or not the description of the tool mentions 'how many' requirements should be used for input. For instance, the description of brainstorming in [9] again states, "A brainstorming session should be focussed on **one** specific function…," thereby indicating that 'one' functional requirement should be used for brainstorming. The entry in the column would therefore be 'one'. However, if the description of the tool does not mention the number of requirements that should be used as an input then the entry in the column would be 'not mentioned'
- *Requirements selection strategy*: Whether or not the description of the tool mentions the strategy used while selecting the requirements is captured in the

fourth column of the protocol. The requirements selection strategy could either be systematic or random. Systematic refers to step by step instruction on how to select the requirements used as input, including the details about the type and number of requirements. If these instructions are not mentioned then the selection strategy is considered to be random. For instance, in describing the four-step approach to analogies as an idea generation tool [56], the author mentions, “1) state the need, 2) generate the analogies …,” but the specific details about how to select the ‘need’ such as whether the ‘need’ is overall need of the project or the need of the sub-system, are not mentioned. In this case, the entry in the column would be ‘random’.

3.4 Discussion

Once the data for both protocols is populated, it is analysed to find patterns to answer the research questions. All observations found are detailed in the following sub-sections.

3.4.1 Findings on the use of Requirements

The first research question aims at understanding how the design-tools taught at different phases of design process in engineering design courses use requirements. Figure 3-1 illustrates the design tools in the planning and clarifying phase of design process. It is important to note that requirements elicitation or any tools involved are not considered as a design tool which makes use of requirements.

As observed, four out of ten design textbooks do not mention design tools in the planning phase of design. Of the remaining six textbooks, most design tool descriptions do not mention the use of requirements.

Examples of the design tools used in the planning phase as discussed in various design text books include Gantt chart, quality function deployment (QFD), product design specification (PDS), requirements list, affinity diagrams and critical path method. Most of these tools are focussed on teaching how to manage the design project and teams. Some of these tools are also focussed on teaching the techniques of understanding the problem at hand and eliciting and documenting the requirements. Therefore, it is obvious that most of these tools will not use requirements since at this stage; the focus is on gathering the requirements.

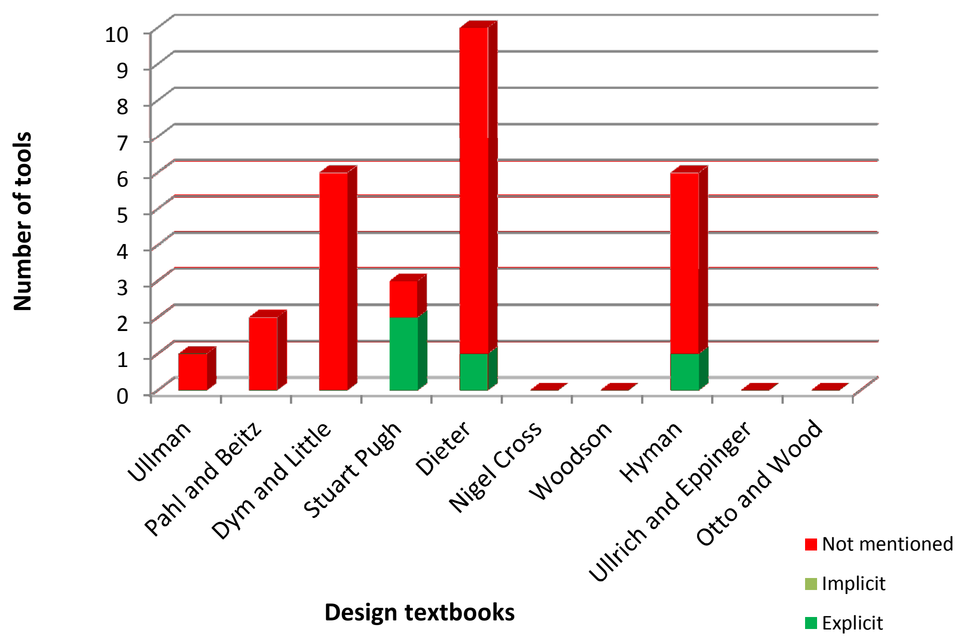


Figure 3-1 Design tools in planning phase

However, there are some tools in the planning phase, such as benchmarking as described in Deiter [53], which mention "identifying the key performance metrics that will be measured and used for comparison," thereby specifying the use of requirements.

Takeaway 1 – *Most design tools in the planning and clarifying stage of the design process* ***do not mention*** *the use of requirements.*

Figure 3-2 illustrates the use of requirements in the design tools in the conceptual phase of the design process. The conceptual design phase consists of generating and selecting the concepts for a given design problem. In generating concepts, requirements must be satisfied or at the very minimum considered, as 80% of the cost of the product is determined at this phase [59]. Therefore, it is essential the tools which aid in the concept generation mention the use of requirements. Further, requirements are also important while selecting amongst the alternative concepts generated as requirements often serve as a selection criteria.

As evidenced from Figure 3-2, most of the design tool descriptions in the conceptual phase mention the use of requirements either explicitly or implicitly.

Most of the idea generation tools fall in the category of design tools that implicitly mention the use of requirements. For example, brainstorming as described in Pahl and Beitz [1]mentions using the problem statement for concept generation but does not explicitly mention using the design requirements for generation of the concepts. On the other hand, brainstorming as described by Ullman [9], mentions using a functional requirement to generate the concepts. Thus, for the same design tool, there is a difference of opinion in using the requirements.

Though not illustrated in the figure, it is found most design tools that explicitly mention the use of requirements fall in the category of concept selection tools. Some examples are decision matrix, Pugh selection matrix, and pair-wise comparison all of which mention the use of criteria to evaluate the alternative design concepts.

What is more interesting to note is there are some concept generation tools which do not mention the use of requirements. For example, idea generation tools such as brain writing and storyboarding as described in Hyman [56] do not mention the use of requirements.

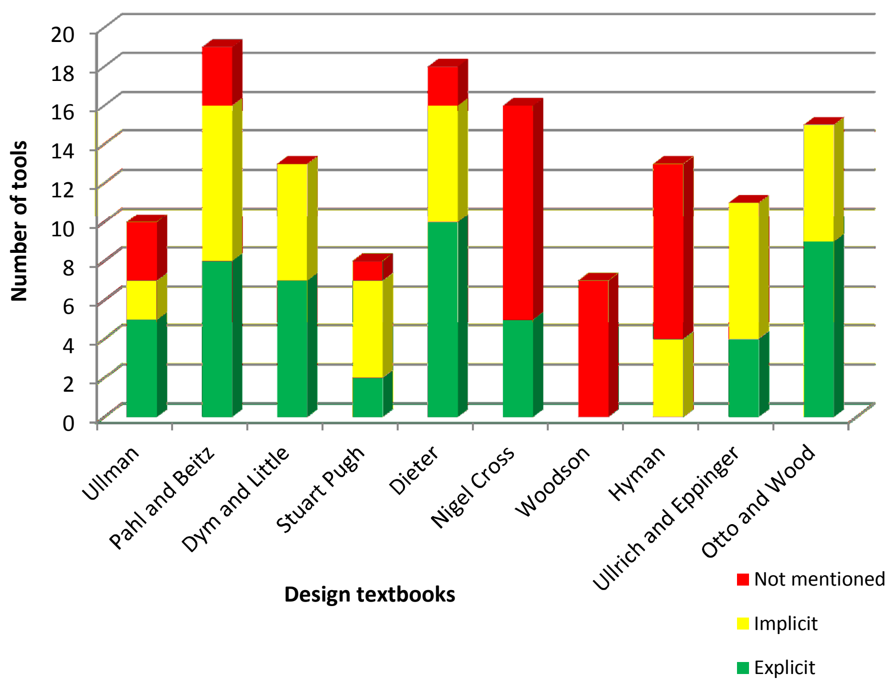


Figure 3-2 Design tools in conceptual phase

> ***Takeaway 2*** *- Most design tools in the conceptual design phase* ***either explicitly or implicitly mention*** *the use of requirements.*

Figure 3-3 illustrates the use of requirements in design tools in the embodiment phase of the design process. Two of the ten design textbooks under investigation do not describe any tools for the embodiment design phase. Of the remaining eight, most design tool descriptions explicitly mention using requirements. Some of the examples of the design tools in the embodiment phase include design for X, failure modes and effect analysis (FMEA), material selection using decision matrix, material selection using Pugh matrix, prototyping, and fault tree analysis.

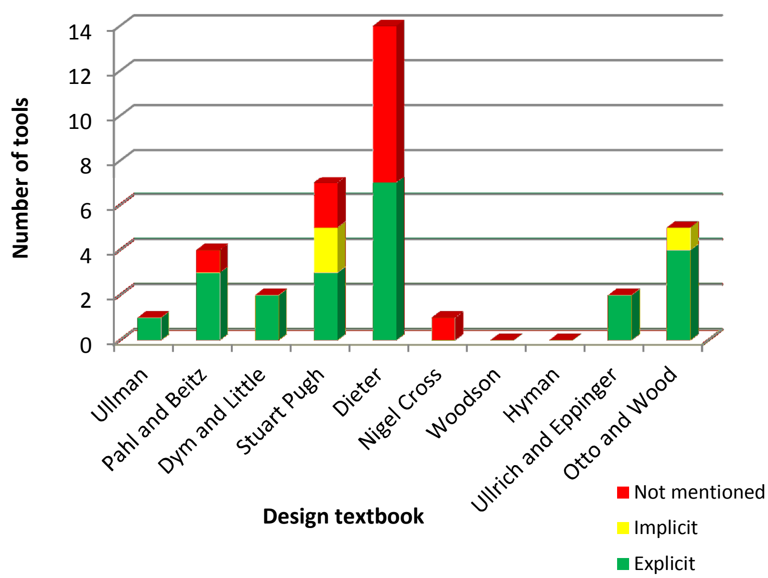


Figure 3-3 Design tools in embodiment phase

***Takeaway 3** – Most design tools in the embodiment design phase **explicitly mention** the use of requirements.*

Figure 3-4 illustrates the use of requirements in design tools in the detail design phase of the design process. Five of the ten design textbooks under investigation, do not introduce any design tools in the detail design phase. Of the remaining five, most of the design tool descriptions do not mention the use of requirements. An illustration of the design tool used in detail design stage, mentioned my most design textbooks, is bill of materials (BOM). Apart from the BOM, design for manufacturing (DFM) and design for assembly (DFA) are also suggested for use in the detail design phase [8]. Of these, only DFM and DFA mention using requirements.

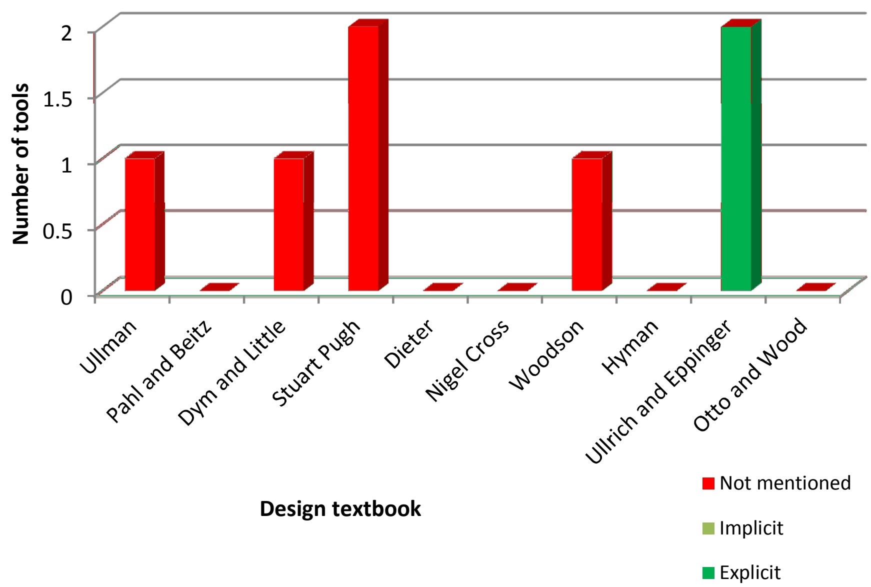


Figure 3-4 Design tools in detail phase

> ***Takeaway 4*** *– Most design tools in the detail design phase* ***do not mention*** *the use of requirements.*

3.4.2 Findings of How Requirements are Described and Explained

The second research question aims at understanding how the use of requirements are described in the design tools, which protocol 1 identified as explicitly or implicitly using requirements. Protocol 2, tabulated in Table 3-3, is used to address the second research question. In doing so, it captures details from the textbooks regarding the design tools' type of requirement, number of requirements, and the requirement selection strategy.

It is important to note that percentage values are used here to draw comparison between different design textbooks, as many of the text varied in the number of design tools presented. From Figure 3-5, it can be observed that in five out of ten design textbooks, the text addresses the type of requirement which should be used in the design tool. Of those which do not provide such information, three of the textbooks lack any information at all regarding how requirements should be used in the design tool, leaving the reader to use their engineering judgement.

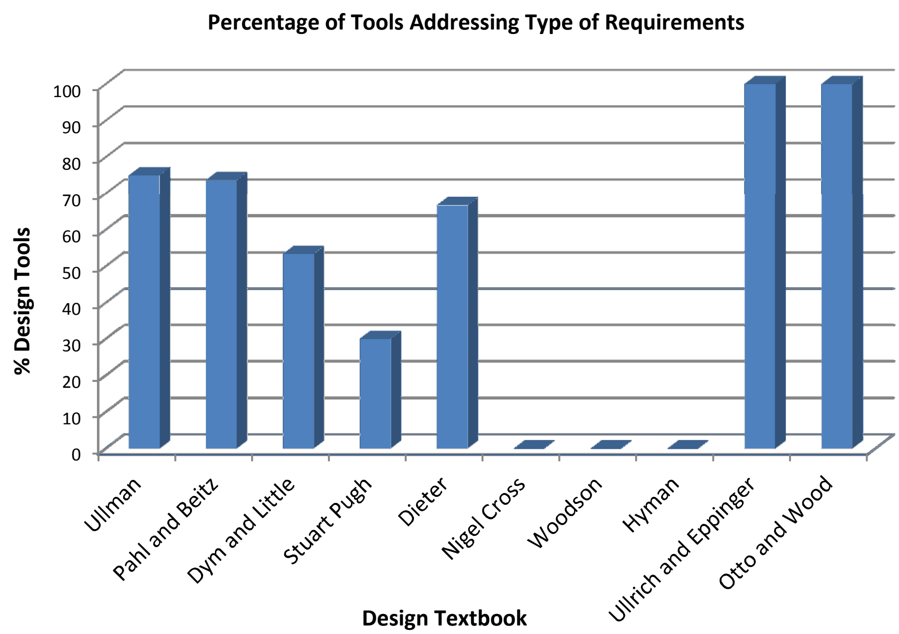


Figure 3-5 Design tools and type of requirements

Alongside the type of requirement that is used, it is important to recognize the number of requirements which should be used. This is of great importance as some design projects may span tens, hundreds, or thousands of requirements. The selection of the number of requirements to use may have an impact on the performance and output of the design tool. Figure 3-6 illustrates the percentage of tools in each textbook analysed which addresses the ‘number of requirements’ used. From the figure, it can be observed in eight out of ten design textbooks, less than 50% of design tools fail to address the number of requirements which should be used, leaving the reader to self-determine the adequate number of requirements.

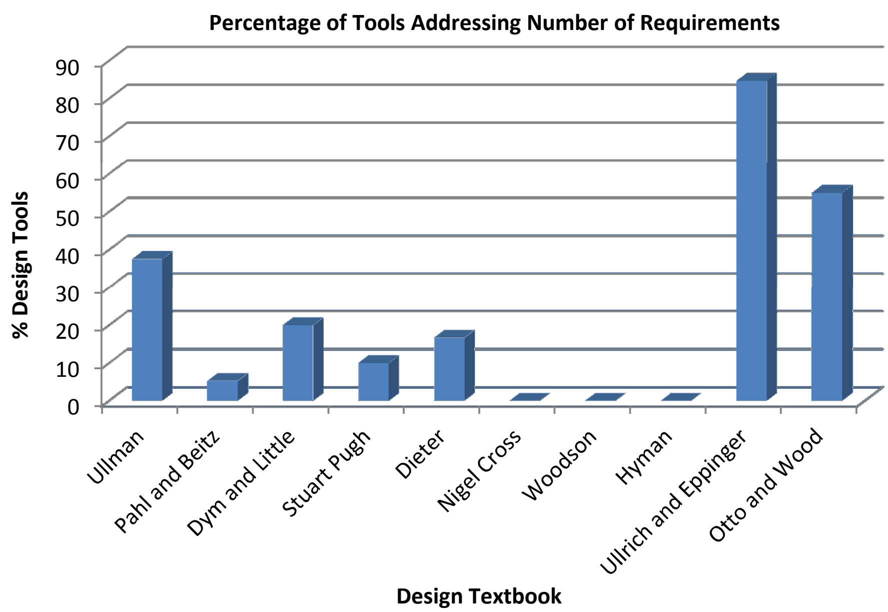


Figure 3-6 Design tool and number of requirements

Figure 3-7 illustrates the percentage of design tools that addresses the selection strategy for requirements. The purpose of a selection strategy is to identify how requirements are selected from the requirements text. The results of the design tool may be influenced by the type of requirements inputted. The results indicated only two out of ten textbooks address the selection strategy of requirements in more than 50 percent of their design tools. This again leaves the reader to assess, under their own judgment, how requirements should be selected for use within the design tool.

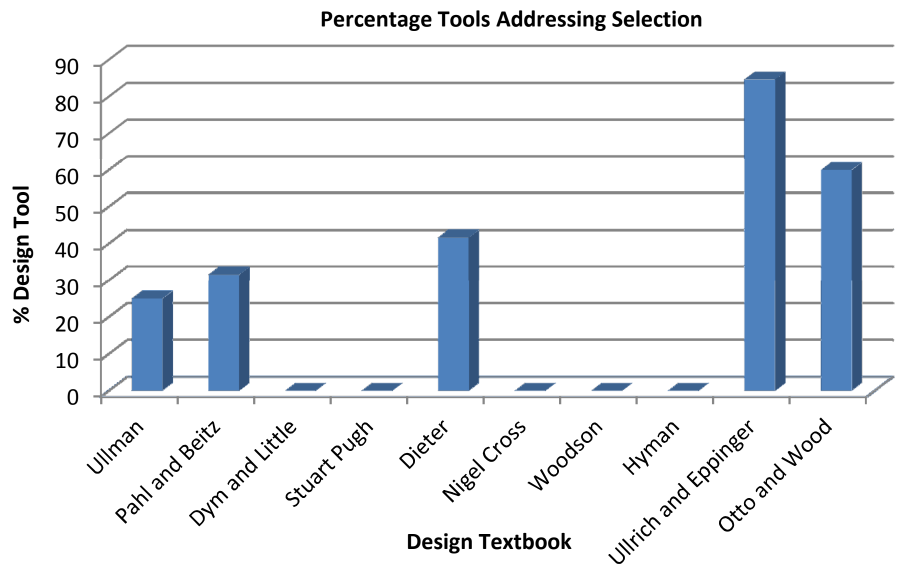


Figure 3-7 Design tool and selection strategy

Protocol 2 was used to collect data addressing the second research question, investigating how well textbooks describe the use of requirements in design tools which make use of them. As seen from the results, in most instances, the reader is left to determine the use of requirements in a design tool based on their own judgment. While this may be justified by the inability to provide specific details due to the vast types of design projects, additional data is needed to assist the reader in using the tools adequately. For instance, two design teams working on similar projects may use the same tool from the same textbook, yet retrieve different outcomes due to the input of requirements (function vs. nonfunctional or varying numbers) into the tool.

Takeaway 5 *–Students are left to assess, under their own judgment, how requirements should be used within design tools.*

3.5 Survey/Interview of Faculty

As previously mentioned, the objective of understanding the current state of requirements education is addressed through two tasks. While reviewing the design textbooks (Task1A) provides a broader perspective on how different design tools incorporate requirements, a survey of faculty (Task1B) may provide a deeper insight into how requirements education is imparted in classrooms. Surveys are typically used as a research method to obtain relevant information from a large sample of population and to demonstrate patterns that are statistically significant [60]. The survey for the purpose of this research was conducted with faculty involved in teaching design courses.

The survey designed for the purpose of this research is divided in three focus areas and a copy is attached Appendix A: Part A of the survey is focused on obtaining general information about the structure of capstone courses at the faculty member's department and is based on the nationwide survey of capstone design courses as reported in [61, 32, 62]. Part B of the survey obtains information about different requirements related activities that students are anticipated to perform while working on a design project. Finally, Part C gathers information about different design tools that students are instructed to use.

A total of five survey responses were collected of which four are from the faculty teaching design courses at various department of Clemson University and one is from a Clemson faculty with the responses based on his experience in teaching design at the University of Arkansas. Again, it may be noted here that while these surveys are not critical to the overall goal of this dissertation, the findings will be used in conjunction with those from the survey of the textbooks to identify critical gaps within requirements education. Thus, five survey responses would suffice for this purpose.

It is evident from the survey responses that most capstone design courses are in existence in their current form for multiple years (minimum 4 to maximum 50 years). Further, the students receive at least one semester of classroom instruction on design before they work on the design projects. This means that classroom instruction plays critical role in the knowledge input for the students.

It is evident from the faculty survey that requirements related activities are fairly spread out throughout the execution of the design project. While the activities such as elicitation, verification and documentation of requirements performed at the start of the project, the activities such as using the requirements for concept generation and evaluation are perform between first quarter to third quarter in the project and finally the activities such as using requirements for concept validation are performed towards the end of the project.

It is interesting to note that while the timeline for other requirements related activities such as elicitation, documentation, use of requirements for concept generation and validation were fairly uniform across different faculty, the timeline for performing requirement change or update was not clear. While some faculty indicated requirements updated occurring throughout the project, other faculty indicated that the update occurred towards the end of the project. This could stem from the different practices followed within different departments but it does indicate that there are ambiguities associated with when requirements should be updated.

While the faculty survey indicates that the requirements related activities are fairly spread through the execution of the design project, the findings from the survey of design textbooks reveal that most tools describing the use of requirements are found in the conceptual design phase. The conceptual design phase is the second phase in the design process following the planning and clarification of the task. In terms of the timeline for project execution for senior design students, this corresponds to the activities performed between first and third quarter. From the survey of the textbook, it is evident that there are fewer tools in planning phase, embodiment phase and detail design phase that describe the use of requirements. Thus, from surveying the textbooks and faculty a significant gap is identified in terms lack of sufficient tools using requirements in each phase of the design process.

Since novice engineers do not possess prior experiences, they rely heavily on the design textbooks and classroom teaching to gain sufficient knowledge for successful execution of the project. Thus, to make sure that the novice designers are gaining the necessary knowledge for successful project execution, it is critical to address the limitations of current education practices.

3.6 Summary of Findings for RQ1-What are we teaching?

This part of research presented a review of the different design textbooks used in undergraduate design education in the United States (US) for developing an insight on how requirements are integrated in the various stages of the design process. This research investigated two focus areas. First, it investigates the mode of reference (implicit, explicit, or not mentioned) to the requirements while using different design tools in different stages of the design process. The findings show that in the planning stage of the design process, only three textbooks have an explicit reference to the use of requirements in the design tools. In the conceptual stage, 90% of the design textbooks have implicit/explicit reference to the use of requirements for the tools they have identified while in the embodiment design stage, 70% of the design textbooks have explicit reference.

In the second part of this research, three factors are explored on how the requirements are described and explained to students while using the design tools. First, it is observed in eight out of ten design textbooks, less than 50% of design tools fail to address the number of requirements that should be used. Second, only 50% of the textbooks indicate the type of requirements to be used. Finally, only two out of ten textbooks address the selection strategy of requirements in more than 50 percent of their design tools. Though the textbooks have recognized the importance of requirements and their proper use and management throughout the design process, their use within specific design tools has been limited, and in most cases, it is left to the reader to digest with a high degree of uncertainty. As requirements specify what the project stakeholders need to satisfy their immediate and end customers, they require greater attention and detail outside of their elicitation and testing.

Thus, the core finding within this research is the lack of rigor given to requirement use within the multiple design tools where they are used. This finding is augmented through the survey of faculty involved in teaching design. The survey of faculty reveals that the requirements related activities are spread throughout the execution of the design project, while from the survey of design textbooks; it is evident that most tools describing the use of requirement are found only in the conceptual phase of design. Thus, there is a significant gap in terms of sufficient tools throughout the design process that would allow the students to use requirements at various phases in design.

After identifying the gaps in the requirements education imparted to the students, Chapter Four focuses on understanding how requirements are used by the students and thus investigates the application of the imparted education.

CHAPTER FOUR: HOW ARE STUDENTS USING REQUIREMENTS –A CASE STUDY

4.1 Study Objective and Overview

The second research goal is to develop an understanding of how are students are currently applying the requirements education imparted to them. With that goal, four specific aspects are investigated:

1) investigating the influence of "delta" in requirements on the level of detail of final solution,
2) investigating the completeness and specificity of elicited requirements,
3) investigating the specificity of requirements met in the final solutions and
4) Investigating evolution of requirements through requirements tracing study

Thus, while it is important to identify critical gaps in the requirements education imparted in the classroom, which is addressed through RQ1, it is also necessary to identify the gaps in application of the imparted education. The findings from both, the study to identify gaps in the dissemination and the application of requirements education, can then be used to make recommendations to overcome the currently limitations.

The case study research method will be used to explore and answer this question. Case study as a research method is used to explore 'why' and 'how' type of research questions [38] and thus this research method is appropriate for answering second research question – *how are students currently using requirements*? Table 4-1 illustrates the anticipated patterns for each sub-question of RQ2.

Table 4-1 Anticipated patterns for Case study

Research Question	Anticipated pattern
RQ2.1 What is the influence of change in number of requirements elicited by individual teams and those provided by the sponsor on level of detail of final solution?	AP2.1 Greater change in number of requirements elicited by individual teams and those provided by sponsor will result to high level of detail of final solution
RQ2.2 What is the completeness and specificity of requirements in initial week and final week?	AP2.2 Teams will have low completeness and specificity in initial week. The completeness and specificity of requirements in final week will increase compared to initial week
RQ2.3 What is the specificity of requirements met in the final solution?	AP2.3 More requirements with high specificity will be met compared to the requirements with low specificity.
RQ2.4 What is the evolution of requirements as evident from requirements tracing?	AP2.4 The requirements will evolve in multiple ways as from initial week to final week.

Pattern matching is often used to study the analyzed data in case study [36, 63]. To improve the qualitative objectivity of the study, counter patterns are also intentionally sought.

RQ2.1 aims at investigating the influence of “delta” in requirements on the level of detail of final solution. The anticipated pattern is that a higher “delta” in requirements will lead to a high level of detail in the final solution. This is based on the assumption that a high delta in requirements essentially means that the team established more requirements compared to those given by the sponsor, which in turn means that they have better understanding of the problem. This will ultimately lead to a high level of detail in the final solution.

Alternately, a high delta in requirements could lead to low or medium level of detail in the final solution. Although a high delta means the team elicited more requirements than those given by the sponsor, if these requirements are identified towards the completion of the project then the students will not have sufficient time to develop the details in the final solution resulting to alternative pattern of low level of detail in the final solution.

RQ2.2 aims at investigating the completeness and specificity of elicited requirements in initial week and final week. The anticipated pattern here is that the teams will have low completeness and specificity of requirements in the initial weeks, while the completeness and specificity will increase in the final week. This is based on the assumption that the students have poor understanding of the problem at hand at the start of the project because they are still fully discovering the problem. Thus, this will result to a poor completeness and specificity in the initial weeks. Towards project completion, the students will have better understanding of the problem thus resulting to increase in the completeness and specificity of the requirements.

Alternately, the completeness and specificity of requirements will not change from initial weeks to final week. From the researcher's personal experience with senior design projects, the students are not given any specific feedback on the completeness and specificity of requirements. This would result to no change in the completeness and specificity of requirements from initial week to final week.

RQ2.3 aims at investigating the specificity of requirements met in the final solution. It is anticipated that more requirements with high specificity will be met compared to requirements with low specificity. Specificity of requirements is essentially a measure of the level of detail in requirements and is measured as a count of number of adjuncts or numerical values. It is anticipated that, the requirements having high specificity as indicated by a high number of adjuncts or numerical values will have sufficient details and thus will be clearly understood by the students. This will then result to more number of high specificity requirements fulfilled in the final solution. Requirements with no adjuncts or numerical values may be too abstract and thus few requirements with no specificity will be met in the final solution.

Alternately, the requirements with high specificity can over constrain the design space and thus pose a challenge while designing a solution that can fulfill them. Thus, an alternative pattern that more number of requirements with low specificity will be met in the final solution is also explored.

RQ2.4 aims at investigating the evolution in requirements through a requirements tracking study. It is anticipated that the requirements will evolve from initial week to final week. As previously mentioned, at the beginning of the design project, the students are exploring and understanding the problem at hand. Towards the end of the project, the student's understanding of the design problem evolves and this will be reflected in the requirements document resulting to multiple changes from initial weeks to final weeks. Further, the changes in the document could also result from the feedback given to the students during the design reviews.

Alternately, it is expected that there will be no change in the requirements from initial weeks to final week. As a part of deliverable for the senior design project the students are not mandated to update the requirements document on weekly basis. Thus, it is possible that the requirement document generated at the beginning of the project is not updated and remains constant. This leads to exploring the alternative pattern that there will be no change in the requirements from initial weeks to final weeks.

Table 4-2 summarizes the alternative patterns explored in this case study.

Table 4-2 Summary of alternative patterns for case study

Research Question	Counter Patterns (CP)
RQ2.1 What is the influence of change in number of requirements elicited by individual teams and those provided by the sponsor on level of detail of final solution?	CP2.1 Greater change in number of requirements elicited by individual teams and those provided by sponsor will result to low level of detail of final solution
RQ2.2 What is the completeness and specificity of requirements in initial week and final week?	CP2.2 Teams will have no change in the completeness and specificity of requirements from initial weeks to final weeks.
RQ2.3 What is the specificity of requirements met in the final solution?	CP2.3 More requirements with low specificity will be met compared to the requirements with high specificity.
RQ2.4 What is the evolution of requirements as evident from requirements tracing?	CP2.4 There will be no change in the requirements from initial week to final week

After discussing the anticipated and alternative patterns for each sub-question for RQ2, section 4.2 describes the senior design class at Clemson University.

4.2 Description of ME-402 class

ME-402 is a senior Mechanical engineering Capstone Design class in Mechanical Engineering department at Clemson University. This course is a three-credit, semester long course of approximately fifteen weeks, except when offered in an abbreviated semester during the summer. As a part of this course, senior mechanical engineering students work in teams of four to five students addressing industry sponsored projects. Each project has three to four teams of students working independently to solve the given design problem. Each project has an advisory committee that consists of two to three faculty, retired industry, or graduate student members. The undergraduate students deliver weekly design review presentations about their progress to the advisory committee, receiving feedback and critiques on the technical, communication, and management aspects of the project. At the beginning of the semester, the students are given one page description of the design problem from the industry sponsor. Additionally, during the first week all teams are given a detailed presentation of the problem by the industry sponsor.

From this point, the multiple student teams assigned to a project work in parallel to develop their own version of problem statement and solutions that address their understanding of the problem. To do this, the students establish the requirements, generate the ideas, and select the final solution based on the requirements. According to the demand of the project, the students also build prototypes to test and validate their concepts.

The students deliver mid-term preliminary design reviews to the sponsors, often with all other project teams in attendance. By the time of these preliminary design reviews, most teams have completed the conceptual and, to some extent, embodied design phases. Most teams have developed a candidate for final solution at this time with the teams spending the remainder of the semester focused on detailing the solutions. Here, the final solution refers to a concept that meets all the constraints established by the team at the beginning of the semester. The final deliverable of the project includes the written report of the proposed solution with complete drawing package and may also include prototypes of the proposed solution. The students are also required to give a final presentation to the advisory committee and industry sponsor.

It may be noted here that since there are multiple teams working independently to solve the same design problem, this serves as an ideal case for studying and comparing requirements elicitation and use. As previously mentioned, there are typically three to four teams working on a design project. At the beginning of the semester, all teams are provided with project description and presentation by the sponsor. From this point onwards, each team works independently to solve the design problem at hand. This results to each team developing their own understanding of the design problem and thus eliciting and updating their own requirements. As each team has their own requirements document independent of other teams, this allows for comparing the document to investigate the similarity and evolution of requirements for each team. If only one team was working on the design project, then this comparison would not be possible. Thus, having multiple teams working independently on the design problem allows for comparing the requirement elicitation and use by the senior design students.

One of the projects from spring 2011 was selected for this case study. There were four teams working on the project and each team had four students. Further, three teams has all males and one team had three males and one female on the team. The teams were graded based on factors such as team work, professionalism, the design process followed, and overall project outcomes. Further, the team and individual grades do not necessarily represent the quality of the project outcome and are therefore not used explicitly for comparing the performance of the four teams under consideration.

The details of the design project under consideration are discussed in section 4.3.

4.3 Design project under consideration

The design project under consideration was completed by the mechanical engineering senior design students in the Spring 2011 semester. The design project was sponsored by Parker Hannifin Corporation. The design problem given to the students was:

"Design and build a system to automatically splice the seals."

There were four teams working independently on this project and each team had four students. Further, the team members consisted of mostly male with an exception of one female on one of the teams. All students were in the final semester of their undergraduate curriculum. The advisory panel consisted of two faculty members and one graduate student. The advisory panel provided feedback to the teams during weekly design reviews.

This project was selected for this study as it is representative of a typical senior design project. While not explicitly measured, the complexity of the design problem, the composition of the design teams and the execution of the process to develop final solution are representative of other senior design project in the mechanical engineering department at Clemson University. Thus, the project was selected for this study. Further, the researcher was a gradvisor on the project and this would facilitate the data collection. At the beginning of the semester, the project sponsor presented the details of the design problem, explicitly stating project requirements to all four teams. After this, all four team worked independently to explore problem and design a solution.

4.4 Data collection

In order to answer the research questions, RQ2.1, RQ2.2, and RQ2.3 it was necessary to collect the requirements elicited by student teams. In a typical senior design curriculum, the students are not required to document the update of requirements on a weekly basis. However, for the purpose of this case study, it was essential to collect the requirements from the student teams each week. In order to facilitate this, the teams were given a 'requirements update sheet' which they were required to update and submit each week. Figure 4-1 shows example requirements update sheet submitted by team C for week 1.

This sheet captures details such as the requirement, date of elicitation or modification, source of requirement (sponsor or team), modification in the requirements, justification for the modification, target value and validation method.

Although the student teams were asked to update and submit the requirements update sheet each week, most teams did not submit one for every week since this was not a course requirement. However, the data has been collected for initial weeks for all teams. Here data from initial week refers to the data from week-1, week-2 or week 3 depending on the student responses.

3	2	1	Sr. No.
Apply controlled amount organic or silicone adhesive to extrusion ends.	Be capable of splicing hollow and solid extrusions.	Grasp extruded circular cross-sections that range in thickness from 0.070'' to 0.250'' and diameters from 2" to 2'.	Requirement
Constraint	Constraint	Constraint	Constraint/ Criteria
Sponsor	Sponsor	Sponsor	Source
Requested by sponsor	Requested by sponsor	Requested by sponsor	Justification
20-Jan	20-Jan	20-Jan	Date of elicitation
Need more info from sponsor	n/a	Described in requirement	Target Value
Calculations or actual results	Calculations or actual results	Calculations or actual results	Verification method
n/a	n/a	n/a	Update (if any)
n/a	n/a	n/a	Condition or Reason for update
n/a	n/a	n/a	Description of Change or Deviation
			Comments (if any)

Figure 4-1 Requirements update sheet

It may be noted that team B did not submit any update sheet for initial weeks, thus the requirements for team B were extracted from the executive summary for week 3. Essentially, team B's executive summary for week 3 had list of elicited requirements. These requirements were used for the analysis. Further, the requirements for final week were extracted from the final report to be consistent across the teams. In addition to the data collected from requirements update sheet, final design report submitted by the teams were used as a data source for analyzing the level of detail of final solution.

Asking the objects of study, which in this case is the senior design students, to collect the information (requirements) used in the analysis has its limitations which are discussed in Section 4.12.

4.5 Data Analysis

After collecting the data, the next step was to analyze the collected data to answer the research questions. As previously mentioned, RQ2 aims at understanding how students currently use requirements. To that end, three different aspects are investigated through this case study: 1) studying the influence of difference in the number of requirements on the level of detail in final solution, 2) studying the completeness and specificity in the requirements across teams working on same project and 3) studying the specificity of requirements met in final solution. Before discussing the findings from these studies, sections 4.5.1, 4.5.2 and 4.5.3 discuss the protocol for analyzing the data to investigate these aspects respectively.

4.5.1 Mapping delta of requirements to level of detail in final solution

RQ2.1 aims to understand the influence of the delta in the number of requirements between the sponsor and the teams on the level of detail of the final solution. At the beginning of the semester, the sponsor provides initial requirements to the teams working on the project. All the teams then work independently to further explore and elicit more requirements. Thus, as they progress in their design process, each team will develop a different number of requirements than what was initially provided by the sponsor. Thus, the 'delta in requirements' refers to the difference in the number of requirements given by the sponsor and that elicited by the teams. Since the researcher was unable to collect the data for all weeks of the project, the comparison for the delta is only made for initial week and the final requirements reported in the final report. After collecting the requirements, the number of requirements for the initial week and the final week were counted and tabulated for each team.

It is hypothesized that team having a greater positive delta should have a better understanding of the design problem and thus have a high level of detail in the final solution. It may be noted here that while looking at the delta, only the change in the number of requirements were considered. Thus, a team that starts with three requirements and ends with three completely different requirements would still have a delta of 3-3=0.

The student teams generate a variety of documents throughout the completion of the project. These include documents such as weekly summaries, weekly design review presentation and mid-term report. However, none of these documents capture the complete details of final solution that are required for this analysis. Thus, final design reports were selected for analysis for this study.

The level of detail of final solution was measured by considering the components of final solution such as description, figures, and engineering. The level of description is measured as count of pages describing the final solution. The level of detail of figures is measured by counting the number of fully, partially and not labeled figures while the level of detail of engineering is measured by considering the number of analysis, experiments and simulations pertaining to the final design. The details of considering level of detail of each component can be found in [64]. The levels of detail of each of the three components were then incorporated to find the level of detail of the final solution [64]. The level of detail of the final solution for all four teams was then mapped to the delta in requirements. The findings from mapping the delta in requirements to level of detail in final solution are discussed in Section 4.6

4.5.2 Completeness and Specificity Study of Requirements

RQ2.2 aims at investigating the completeness and specificity of requirements within and across teams. While completeness of a requirement is measured by considering components such as system (subject), necessity (modal), behavior or characteristics and condition (verb phrase), the specificity of a requirement is measured by considering the number of adjuncts and numerical values within a requirement [17]. Thus, in order to measure the completeness and specificity of requirements, it was essential to parse the requirements into components.

For each team, the requirements from the initial weeks and the final week were parsed and the number of adjuncts and numerical values were counted and tabulated. Table 4-3 shows a snap shot of the evaluation table created to measure the completeness and specificity of requirements.

Table 4-3 Evaluation table for completeness and specificity of requirements

Requirement	Parsing for completeness			Specificity	
	Subject (System)	Modal (Necessity)	Verb phrase (Behavior and condition)	# of adjuncts	# of numerical values
The design must cool the spliced o-rings in less than or equal to 3 minutes	Design	Must	Cool the spliced o-ring in less than or equal to 3 minutes	2	1
Count the number of parts	0	0	Count number of parts	0	0

The first column indicates the requirement elicited by the team. Column 2 is the completeness column and shows parsing for completeness. Each requirement elicited by a team was parsed into “subject”, “modal” and “verb phrase” to measure completeness. If a requirement did not have a component, then ‘0’ was entered in the respective column. For instance, the requirement “*count the number of parts*” does not have a subject and modal and thus ‘0’ was entered in the columns for both the subject and the modal for this requirement. Column 3 represents specificity column recording the number of adjuncts and numerical values in order to measure specificity. Again, if a requirement did not have any adjuncts or numerical value, ‘0’ was entered in the column.

It may be noted that this protocol for analyzing the completeness and specificity of requirements is established in [17] and was adopted in this thesis for manual coding.

4.5.3 Requirements met in final solution

The goal of RQ2.3 is to investigate whether or not more requirements with high specificity are met in the final solution as compared to the requirements with low specificity. Specificity of a requirement is measured by using the number of adjuncts and numerical values as discussed above. Thus, the higher the number of adjuncts and/or numerical values, the higher the specificity of requirement is measured [17].

Each team submitted a final report as the penultimate project deliverable. In order to identify the requirements met, the final reports for all four teams were read thoroughly. The requirements that were explicitly met were counted and recorded. It may be noted that the mapping of requirements to final design solutions established in [64] considers whether or not a requirement was addressed, however for this research whether or not a requirement was explicitly met was considered. A requirement was considered as "met" if the report had a description, figure, or analysis that clearly suggested that the requirement was met. The requirements for which there was no evidence suggesting that it was met were considered as "not met". Subjective speculation about whether a requirement might have been met, but not fully discussed, was avoided to ensure as objective analysis of the final solution as possible. Further, only the details pertaining to final solution were considered while analyzing for requirement met.

After recording the number of requirements met and not met, comparison was made against the specificity of requirements to investigate whether or not more number of requirements with high specificity is met as compared to requirements with low specificity. The findings from this study are discussed in Section 4.9.

4.5.4 Requirements tracing protocol

The goal of conducting a requirements tracing study was to explore the evolution of requirements and identify the changes in individual requirements from initial weeks to final week. Further, three different evolution aspects were investigated in this study - 1)number of addition, deletions, changes and no changes, 2)identifying changes in natural language requirement elements (system, necessity, behavior, object and condition) and 3)identifying change types as per the change taxonomy established in [16, 65]. In order to study this evolution, requirements tracking sheet was created for each team and each requirement was traced as it evolved.

Figure 4-2 illustrates a snap-shot of requirements tracking sheet for team A. The first row represents the requirement code for initial week and final week. Thus '7' indicates requirement-7 for initial week and '7F' represents requirement-7 for final week. Each requirement in the initial week was traced down to final week in similar manner. This tracing was done manually by identifying similar requirements.

It may be noted that from initial week to final week, some requirements were deleted and new requirements were added. Thus the change type for these requirements was identified as 'addition' or 'deletion'. For example, in Figure 4-2, there was no requirement corresponding to requirement '6F' in initial requirement document and thus the change type was identified as 'addition'.

	Initial Week	Final Week	Initial Week	Final Week	Final Week
	7	7F	9	9F	6F
PARKER TEAM A	Cool spliced rings	The design must cool the spliced rings in less than or equal to 3 minutes	Count the number of parts	The design must count the number of parts produced	The design must prevent breakdown in elastomer properties by ensuring that o-rings are not heated beyond 350 F
Subject	0	The design	0	The design	The design
Modal	0	must	0	must	must
Main Verb	Cool (T)	cool (T)	Count (T)	count (T)	prevent(T)
Direct Object (DO), NA, complement (CO)	spliced rings (DO)	the spliced rings(DO)	the number of parts (DO)	the number of parts produced (DO)	breakdown in elastomer properties (DO)
Adjunct	0	in less than, or equal to 3 minutes	0	0	by ensuring that o-rings are not heated, beyond 350F
Change Type	1. Introduction of new system 2. Importance 3. Specificity		1. Introduction of new system 2. Importance		Addition
NL requirement element	1. System 2. Necessity 3. Condition		1. System 2. Necessity		N/A

Figure 4-2 Example requirements tracking sheet for team A

Similarly, if a requirement in initial requirement document did not end up in the final requirements document then the change type was recoded as a 'deletion'. If a requirement was changed from initial week to final week, it was counted as 'changed' while if the requirement did not change at all it was counted as ‘no change’. Thus, the number of addition, deletions, change and no change were counted and recorded in a summary table as shown in Table 4-4.

Table 4-4 Summary of addition, deletion, no change and change

	Parker A	Parker B	Parker C	Parker D
Additions	1	2	3	16
Deletions	0	3	10	0
No change	0	2	0	0
Change	15	11	10	8
Total initial	16	16	20	8
Total final	16	15	14	24

Next step was to identify specific change in requirement in terms of changes in natural language requirement elements and then identifying specific change types based on taxonomy established in [16]. To that end, first, the requirement in initial week and final week was parsed into the components- system (subject), necessity (modal), behavior/characteristic (main verb), object and condition (complement or adjunct) [17, 65]. A comparison was made between initial week and final week components to identify which of the components changed. So for instance, consider the following requirement pair for team A:

(Initial week)7-Cool the spliced ring.

(Final week)7F –The design must cool the spliced ring in less than or equal to three minutes.

Here, in requirement 7F, the subject “design”, modal “must” and adjuncts “in less than or equal to three minutes” were added. These were thus recorded as change in system, necessasity and condition as shown in Figure 4-2. Further it may be noted that this was only recorded as a change, whether it was addition, deletion or modification was not considered for the purpose of this study. After completing similar coding for all requirement pairs for all teams, number of each element change were counted and recorded in a summary table as shown in Table 4-5.

Table 4-5 Summary of changes in requirement elements

	Parker A	Parker B	Parker C	Parker D
System	15	4	10	8
Necessity	6	3	10	0
Behavior	6	6	2	3
Object	5	2	3	4
Condition	11	8	2	4
Total number of changes	43	23	27	19

After identifying the change in natural language requirement elements, the next step was to identify the change types based on the taxonomy of change types established in [16]. The definitions and examples of each change type provided in the taxonomy were used to identify the change type for each requirement pair.

While a complete list of all change types with examples can be found in [16], Table 4-6 provides explanation of change types identified by examining the requirements for this study.

Further, a requirement pair could have multiple change types. So for instance consider the following requirement pair for team A:

(Initial week)7-Cool the spliced ring.

(Final week)7F –The design must cool the spliced ring in less than or equal to three minutes.

Table 4-6 Description of change types, adapted from [16, 65]

Change type	Reference	Description
Introduction of new system	[66, 67, 68, 69]	Identified as change in the system of the requirement. this could be addition of new component or system
Consistency	[66, 70]	Identified as change in vocabulary, units or terminology
Importance	[70]	Identified as change in the necessity of requirement
Specificity	[70]	Identified as change in the level of detail of a requirement
Application	[67]	Identified as change in the application of part or system
Measurability/testing	[70]	Identified as change in the measurability of a requirement
Withdrawal of system	[66, 67, 68, 69]	Identified as change in the system of the requirement. This could be a removal of system or component
Merging	[66]	Identified as change when two or more requirements are merged into single requirement
Associated user	[71]	Identified as change when the individuals associated with the requirement change
Splitting	[66]	Identified as change when one compound requirement is split into two or more requirements
Scope change	[66, 68]	Identified as change when the scope or focus of a requirement changes

For this requirement pair, three change types were identified. First, in requirement 7F the system “design” was added and thus it was identified as change type introduction of new system. Second, the necessity “must” was added and thus identified as change type importance. Finally, the condition “in less than or equal to three minutes” was added and this was identified as change type specificity. The three change types were identified for requirement pair 7 and 7F.

The change types for all requirement pairs for all teams were identified in this manner and a summary table was created by counting the number of each change type for each team as shown in Table 4-7.

Table 4-7 Summary of change types

	Parker A	Parker B	Parker C	Parker D
Introduction of new system	15	0	10	0
Application	3	2	0	1
Specificity	7	4	1	2
Merging	1	0	0	0
Importance	6	3	10	0
Consistency	4	7	1	8
Measurability/Testing	2	1	0	1
Withdrawl of system	0	2	0	0
Associated user	0	1	0	0
Splitting	0	0	1	0
Scope change	0	0	1	0

This section describes the coding protocols for the studying the requirements evolution. The numbers summarized in Table 4-4, Table 4-5 and Table 4-7 were then used for requirements tracing study. The findings are discussed in section 4.10.

4.6 Findings from mapping delta in requirements to level of detail of final report

RQ2.1 aims at investigating the influence of delta in requirements on the level of detail of final solution. The anticipated pattern here is a team with higher positive delta will have high level of detail in the final solution. This is based on the hypothesis that a team having higher number of requirement has a better understanding of the problem and thus would have high detail in the final solution. After counting the number of requirements elicited by each team in initial week and final week, these numbers were compared with the sponsor. Before discussing the findings from mapping the delta in requirements to level of detail of final solution, Table 4-8 shows the summary of number of requirements for sponsor and for each team for initial week and final week.

Table 4-8 Summary of number of requirements in initial week and final week

	Sponsor	Team A	Team B	Team C	Team D
Number of requirements in initial week	13	16	16	20	8
Number of requirements in final week	13	16	15	14	24

The "delta" in requirements was calculated as the difference of team requirements and sponsor requirements for initial week and final week. Thus "delta" in requirements for initial week for team A would be (16-13 = 3). Further, a positive "delta" indicates that the team had more number of requirements elicited compared to those given by the sponsor. A negative "delta" indicates that the team had fewer requirements elicited compared to those given by the sponsor.

Figure 4-3 illustrates the findings from calculating delta compared to sponsor for each team for initial week and final week. It can be observed that team A has consistent delta compared to sponsor for initial week and final week. This means that team A had the same number of requirements in initial week and final week. While team B has a positive delta for both initial week and final week, the difference was higher in initial week compared to final week. Thus team B had more requirements in initial week than final week.

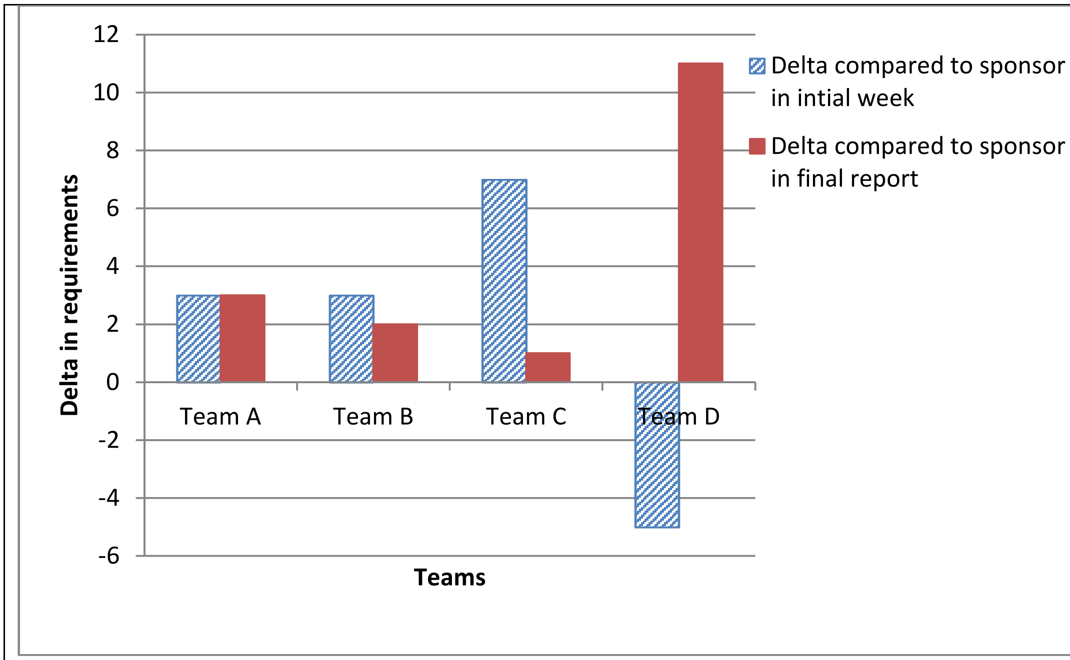


Figure 4-3 Delta in requirements for initial week and final week

Further, team C has a high positive delta in initial week. This indicates that team C had elicited significantly high number of requirements than those provided by the sponsor in initial week. Although team C has a positive delta in final week, indicating that it has more requirements than those provided by the sponsor, the number of requirements in final week has reduced to fourteen compared to twenty requirements in initial week thus resulting to a low positive delta in final week.

It is interesting to note that of the four teams; only team D has a negative delta in requirements for initial week. This could potentially mean that the team failed to identify the requirements explicitly established by the sponsor. However in the final week, team D has a high positive delta in addition to having the maximum number of requirements elicited when compared with other teams.

These observations suggest that while all teams are working on the same project, based on the number of elicited requirements in initial and final week and comparing the delta with sponsor, they seem to have a different understanding of the problem at hand. It is expected that the teams have different understanding of the problem towards the project completion because they are working independently to develop solutions. But in the initial weeks, it is expected that the teams have similar understanding of the problem as they have a common source of input –the sponsor. Solely based on the number of requirements, teams A, B and C had more requirements than those provided by the sponsor in initial week, but team D failed to identify the requirements established by the sponsor as represented by a negative delta. This raises a question on the requirements elicitation practice taught to and followed by the students. It may be noted that the students were had the same pre-requisite design class but they were taught by different instructors. While the instruction styles may be different, it would be expected that the students are learning similar basic tools of design. The difference in requirements elicitation pattern between student teams with team C having too many requirements and team D too few requirements in initial week suggest the contrary.

After studying the delta in requirements for all teams, the next step was to map the delta in requirements to the level of detail in the final solution. Figure 4-4 illustrates the findings from mapping the delta in requirements to level of detail of final solution.

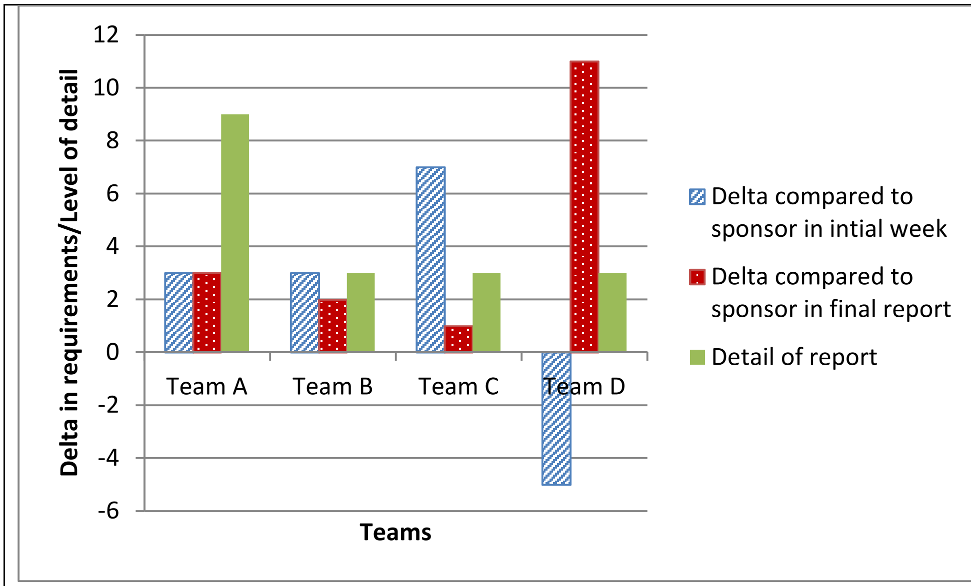


Figure 4-4 Mapping delta in requirements to level of detail of final solution

It can be observed that team A that has consistent delta in initial week and final week and it has a high level of detail in the final solution. While the remaining three teams, B, C and D that have a change in delta from initial week to final week have medium level of detail in final solution.

A consistent delta in requirements could be an indication that students have consistent understanding of the problem in initial week and final week. A consistent understanding of the problem ultimately leads to a high level of detail in the final solution. On the other hand, change in the delta in requirements from initial week to final week may be suggestive of change in the understanding of the problem from initial week to final week resulting to a medium level of detail in the final solution. As with all design projects, the capstone design projects are time bound. While it is desired that the understanding of the problem evolves with the progress in the design project, missing out on requirements early on in the project could potentially affect the detail of the final solution. If the requirements are identified too late in the process, the teams may not have sufficient time to address those requirements.

Further, it is desired that the students' understanding of problem evolves as they progress in the design project and thus reflected through increase in the number of requirements from initial week to final week. However, in this case, for two out of four teams (teams B and C), the number of requirements in final week decreased as compared to initial week. This could mean that these teams had over constrained problem to begin with and as they progressed they removed the unnecessary constraints as indicated by the decrease in the number of requirements.

Thus these findings suggest that there is some ambiguity in the requirement elicitation practice followed by the students. One reason for this could be lack of appropriate design tools that allow students to systematically elicit and document requirements.

After discussing the findings from mapping delta in requirements to level of detail in final solution, section 4.7 discusses the findings from completeness study.

4.7 Findings from completeness study

Completeness of a requirement is measured by considering components such as system represented by subject, necessity represented by modal, behavior/characteristics and condition represented by verb phrase [25, 17, 16]. After parsing the requirements of the initial weeks and the final week for each team, a summary table was created by counting the requirements that are complete and missing one or more of the components.

The summary table for initial week is shown in Table 4-9 while Table 4-10 shows the summary table for final week. Further, each requirement had multiple missing components and so the total number of requirements shown in the last row is not equal to the sum of each column.

Table 4-9 Completeness summary for initial week

Initial week	Team A	Team B	Team C	Team D
Complete	1	16	0	8
Missing Subject	14	0	20	0
Missing Modal	11	0	20	0
Missing Verb Phrase	0	0	0	0
Missing Subject and Modal	10	0	20	0
Total Number of Requirements	16	16	20	8

Table 4-10 Completeness summary for final week

Final Week	Team A	Team B	Team C	Team D
Complete	10	10	14	24
Missing Subject	0	1	0	0
Missing Modal	6	4	0	0
Missing Verb Phrase	0	0	0	0
Missing Subject and Modal	0	0	0	0
Total Number of Requirements	16	15	14	24

Figure 4-5 illustrates the completeness of requirements for all four teams in the initial weeks. From Figure 4-5 it can be observed that for teams B and D, all the elicited requirements are complete. Thus, all the requirements have the necessary components such as subject, modal, and verb phrase. For team A, only one of the sixteen elicited requirements were complete; while for team C none of the elicited requirements were complete.

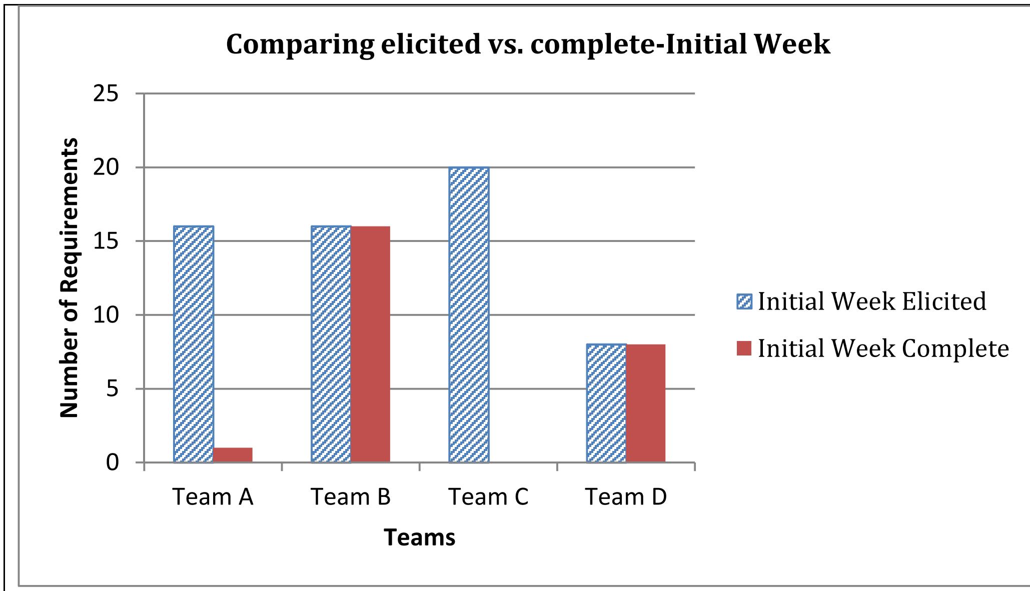


Figure 4-5 Completeness of requirements in initial week

It was necessary to further investigate which components of requirements were missing in the requirements that were incomplete. From Figure 4-6, it can be observed that for teams with incomplete requirements most of the incompleteness is related to missing subjects and modals; as is the case for teams A and C who are are missing subject and modal. The subject represents the system associated with the requirement; while the modal represents the necessity or criticality of a requirement, distinguishing between constraints and criteria. It is interesting to note that none of the requirements are missing verb phrase.

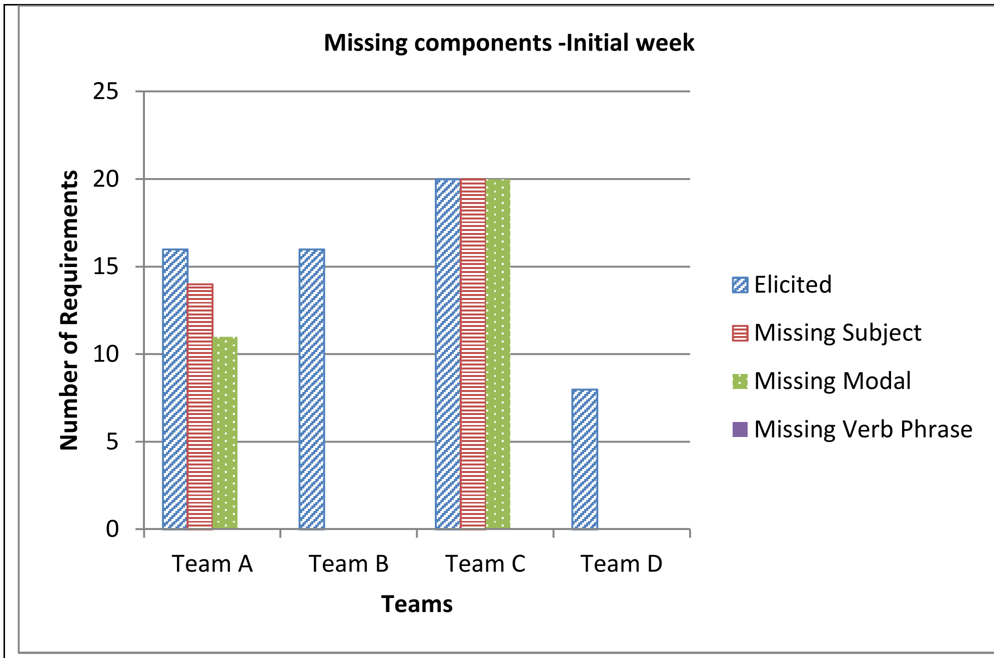


Figure 4-6 Missing elements – initial week

Figure 4-7 illustrates the requirements completeness levels in the final week for all the teams. In the initial weeks, only teams B and D had complete requirements, and team A had only one complete requirement. While the completeness of requirements for team B decreased from the initial weeks to the final week, teams A, C, and D were able to increase the completeness of their requirements from the initial weeks to the final week, as can be observed in Figure 4-7. This is latter pattern was expected as the teams would more likely have a better understanding of requirements towards the project completion.

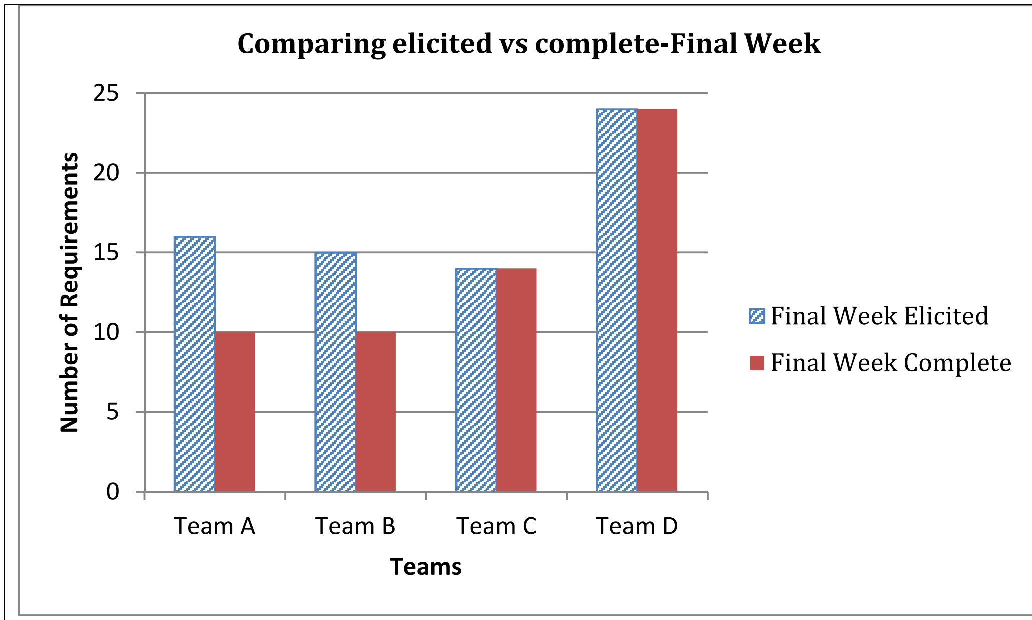


Figure 4-7 Completeness of requirement in final week

Figure 4-8 illustrates the missing components in requirements for all teams in final week. It can be observed that while in the initial week team A had fourteen requirements missing subject, in final week it has none of the requirements missing the subject. Further in the initial weeks, team A also had eleven requirements missing the modals. In final week, this reduced to only six requirements missing the modal. On the other hand, team B had all requirements complete in the initial weeks but in the final week, it has one requirement missing the subject and four requirements missing the modal.

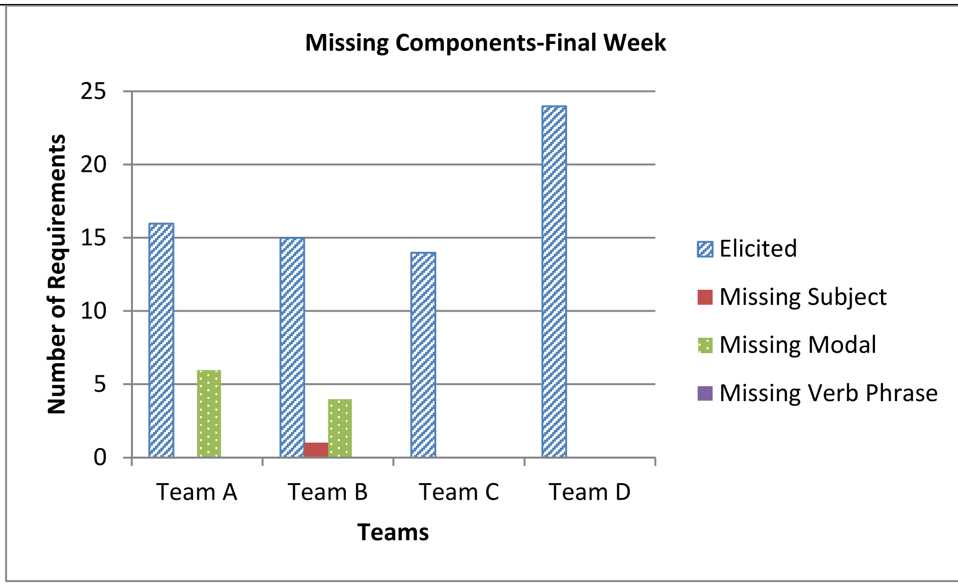


Figure 4-8 Missing components in final week

From above findings it can be inferred that the overall completeness in requirements increase from initial week to final week. While there is significant number of requirements missing either subject or modal in initial week, the number of requirements missing subject and modal decreases in final week.

It is essential to have a complete requirement statement in order to identify what is being designed (indicated by subject), whether a requirement is a constraint or criteria (indicated by modal) and the detail of what the system must do (indicated by verb phrase) [17]. If the requirement is missing one of these components, it could lead to potential ambiguity while interpreting and meeting the requirement. So for instance, one of the requirements in the initial week document for team A is "*count the number of parts*". This requirement is missing the subject and modal. It is therefore ambiguous whether this requirement is a constraint or criteria. Further it is not known 'what must count part'.

Therefore it is ambiguous whether the goal is to design a system or a component to count parts. Alternatively, requirement could also be interpreted as having an operator to count the number of parts. Thus, incompleteness in requirement leads to ambiguity in interpreting and addressing the requirement.

While the requirement may have poor completeness in initial weeks as the requirement document is still evolving, it is desired that the completeness increases towards final weeks to mitigate the ambiguity.

The findings of the case study reveal that while the completeness of requirements increases from initial weeks to final week, there are still few requirements missing important components such as subject or modal as observed for teams A and B. Based on the researcher's experience with advising senior design projects, the students are not given specific feedback about the completeness of the requirement document or the requirements themselves. Thus the increase in the completeness could solely be a result of general feedback or evolution in students understanding of the design problem at hand. Based on these findings, it is recommended that the student should be given specific feedback about the completion status of the requirements to mitigate the potential dangers of having incomplete requirements.

4.8 Findings from Specificity study

Specificity of requirements is a measure of the level of detail within a requirement [17]. This detail could be about the system component or the behavior or characteristics of the system [17]. It is desirable that the requirement has a high level of specificity to minimize the ambiguities associated with it [17]. Specificity of engineering requirement is measured by counting the number of adjuncts and numerical values within a requirement [17].

If the requirements have poor specificity and thus are more abstract, it could lead to failure in addressing the requirements as the interpretation of the requirement would be ambiguous. Therefore, it becomes necessary to investigate the level of specificity within requirements elicited by novice designers. The anticipated pattern here is that the teams will have low specificity of requirements in initial week. This will be indicated by more requirements with less adjuncts and less numerical values in initial week. Further, it is anticipated that the students will have high specificity towards the final week as they will have better understanding of the problem then. An indication of this will be increase in the number of requirements with more number of adjuncts and numerical values.

While the change in specificity for each requirement is out of scope for this research, the findings from the specificity study of initial and final requirement documents are discussed in Sections 4.8.1 and 4.8.2.

4.8.1 Specificity as count of adjuncts

After tabulating the number of adjuncts for each requirement for each team, a summary table was created by counting the requirements with zero, one, two and three adjuncts. None of the requirement had more than three adjuncts and thus the number was limited to three.

Figure 4-9 illustrates the comparison of requirements with 0 adjuncts for initial week and final week for all teams.

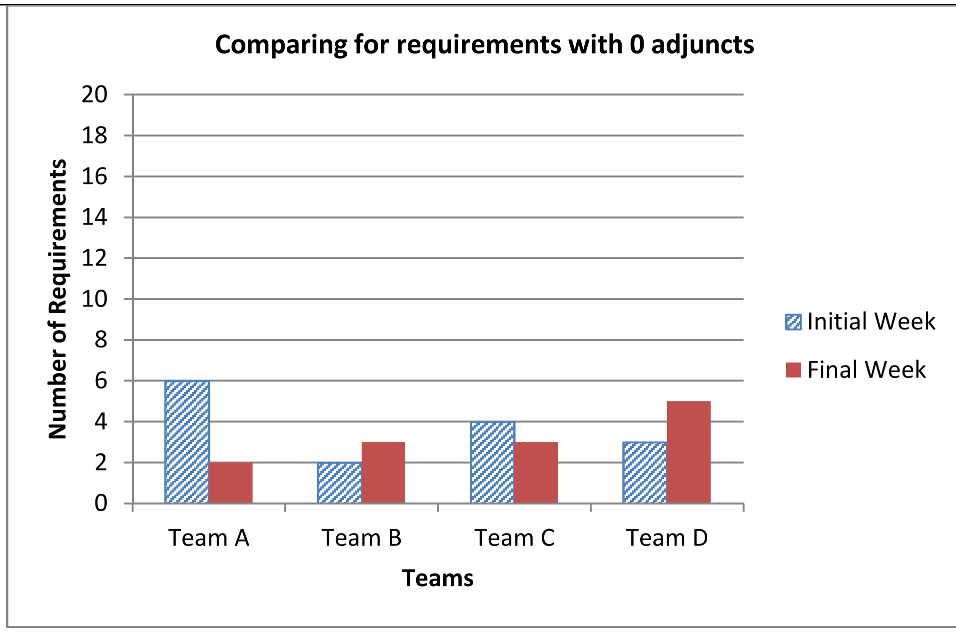


Figure 4-9 Comparing for requirements with 0 adjuncts for initial and final week

From Figure 4-9, it can be observed that for teams A and C, the number of requirements with zero adjuncts have decreased from initial week to final week. For teams B and D there is increase in the number of requirements with zero adjuncts.

Figure 4-10 illustrates the comparison of requirements with one or more than one adjuncts for initial week and final week for all teams.

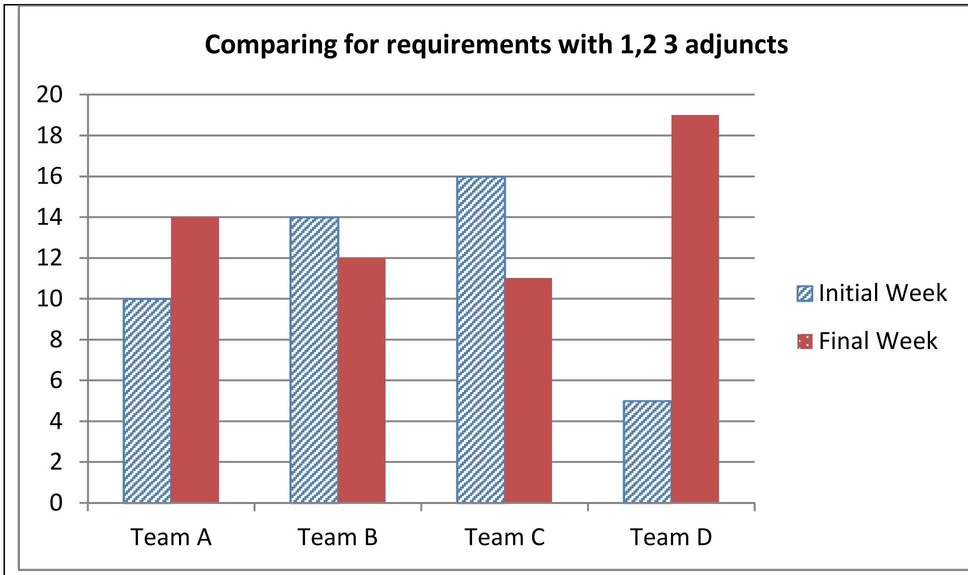
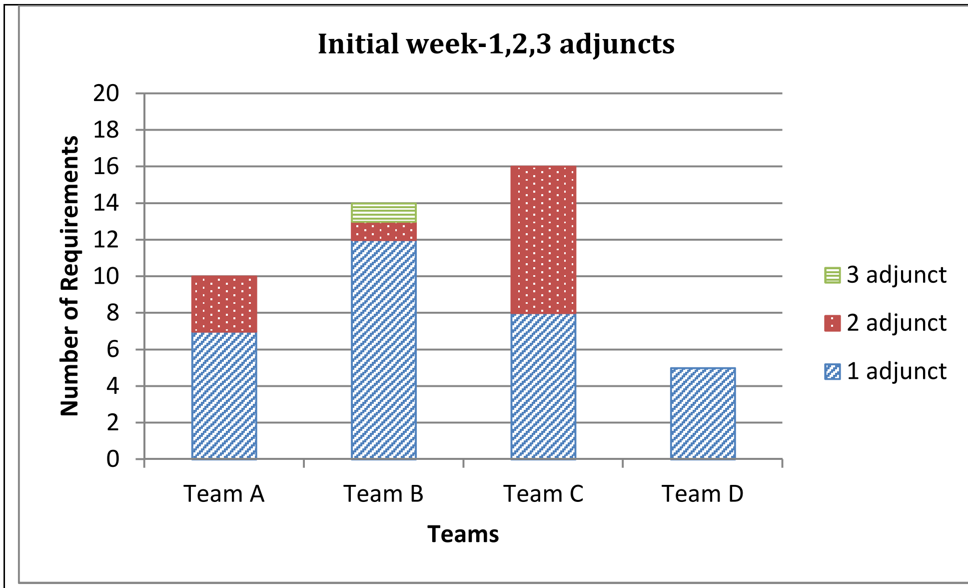


Figure 4-10 Comparing for requirements with 1 or more adjuncts for initial week and final week

From Figure 4-10, it can be observed that for teams A and D there is increase in the number of requirements one or more than one adjunct. There is decrease in the number of requirements with one or more than one adjunct for teams B and C.

Further investigating the requirements with one or more than one adjunct, Figure 4-11 illustrates the comparison of requirements with one, two and three adjuncts for initial week and final week.



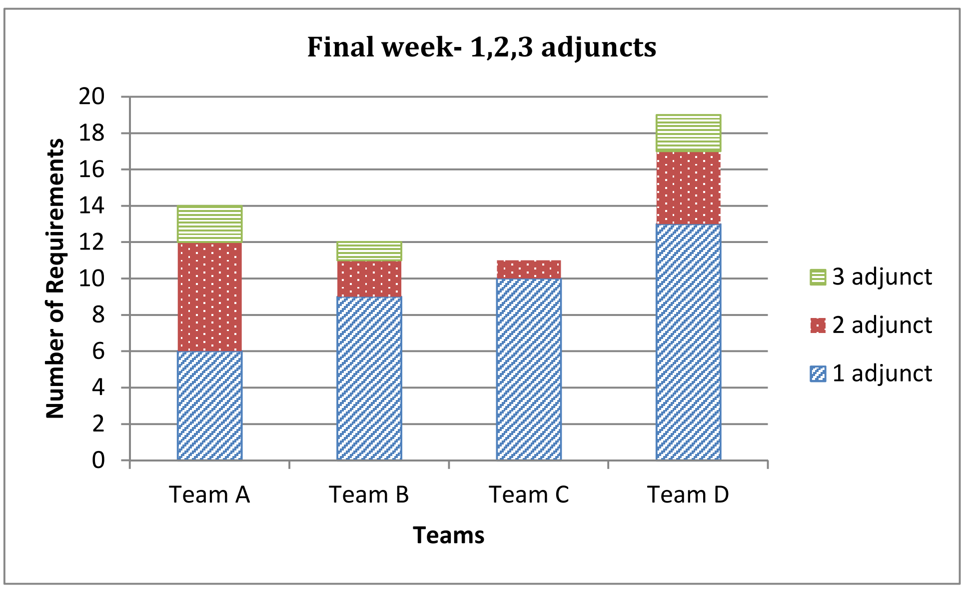


Figure 4-11 Comparing requirements with 1,2,3 adjuncts for initial week and final week

Comparing the graphs for initial week and final week from Figure 4-11, it can be observed that for the teams C and D there is increase in the number of requirements with one adjunct from initial week to final week. The number of requirements with one adjunct decreased for teams A and B.

For teams A, B and D, there is increase in the number of requirements with two adjuncts from initial week to final week. Further, it is interesting to note that team D did not have any requirement with two adjuncts in initial week, while in final week it had four requirements with two adjuncts. For team C, the number of requirements with two adjuncts decreased considerably.

In initial week, only team B had one requirement with three adjuncts. None of the other teams had requirements with three adjuncts. While in the final week, three out of four teams (A, B and D) had requirements with three adjuncts. Thus there was an increase in the number of requirements with three adjuncts.

Thus, the findings suggest that the number of requirements with zero adjuncts decrease and the number of requirements with one, two and three adjuncts increase from initial week to final week. This leads to the conclusion that there is increase in the specificity of requirements measured as the number of adjuncts from initial week to final week.

Specificity of a requirement describes the level of detail within a requirement. An increase in the level of detail of requirements is expected as the students approach the project completion. This is reflected by increase in specificity with increase in number of adjuncts. This is due to the increased understanding of the design problem. Specificity of requirement can thus be used as an internal measure of project completion. Students can be taught to identify the specificity in requirements. If the students are approaching project completion week and they have poor specificity in the requirements, this could essentially mean that they have failed to identify specific details of requirements. Failing to identify necessary details in requirement can then jeopardize successful completion of the project.

For instance, for team C, one of the initial requirements statements is "*Cool the spliced rings*". This requirement has zero adjuncts. The requirement is also missing specific details such as 'what' should cool the rings and in 'what time' the rings should be cooled. However, these details are added in the final week document and the requirement evolves as "***The design*** *must cool the spliced rings in* ***less than or equal to 3 minutes***". In this requirement the subject 'design' and two adjuncts 'less than' and 'or equal to 3 minutes' were added increasing the specificity of the requirement. Had the students not identified these details, there would be ambiguities associated with meeting this requirement. Thus, to mitigate the ambiguities, it is desired that the specificity in requirements as count of adjuncts increases from initial week to final week of the project.

Next, the specificity as a count of numerical values is investigated.

4.8.2 Specificity as count of numerical values

In addition to the number of adjuncts, the number of numerical values is also a measure of specificity of a requirement. While adjuncts specify the detail in a requirement, numerical values add a quantitative element to the requirement [17]. After investigating the specificity of requirements as a count of adjuncts, the next step was to measure the specificity by considering the number of numerical values within a requirement.

After counting the number of numerical values for each requirement, a summary table was created by counting the number of requirement with zero, one, two, three and four numerical values. None of the teams had more than four numerical values within a single requirement and thus the number was limited to four.

Figure 4-12 illustrates the comparison of requirements with zero numerical value for initial week and final week.

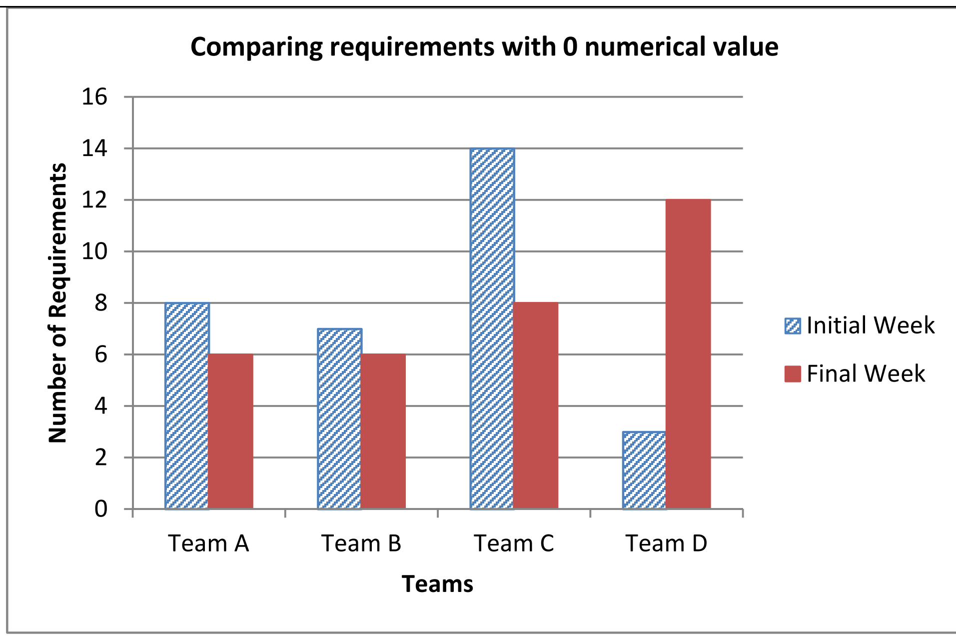


Figure 4-12 Comparing requirements with 0 numerical values for initial week and final week

From Figure 4-12, it can be observed that for teams A, B and C, the number of requirements with no numerical values decreased from initial week to final week. For team D, the number of requirements with no numerical value increased. However, this could stem from the fact that team D added sixteen new requirements in the final week document. Figure 4-13 shows the comparison of requirements with one or more than one numerical value.

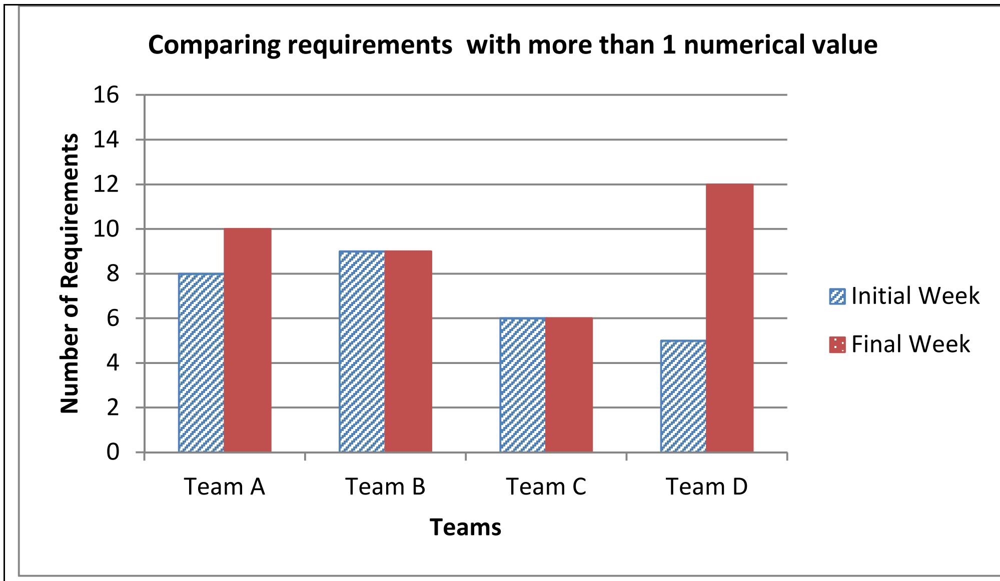


Figure 4-13 Comparing requirements with 1 or more numerical values for initial week and final week

While the number of requirements with one or more than one numerical value increased from initial week to final week for teams A and D, there was no change in the number of requirements for teams B and C. Figure 4-14 illustrates the comparison of requirements with 1, 2, 3 and 4 numerical values for initial week and final week.

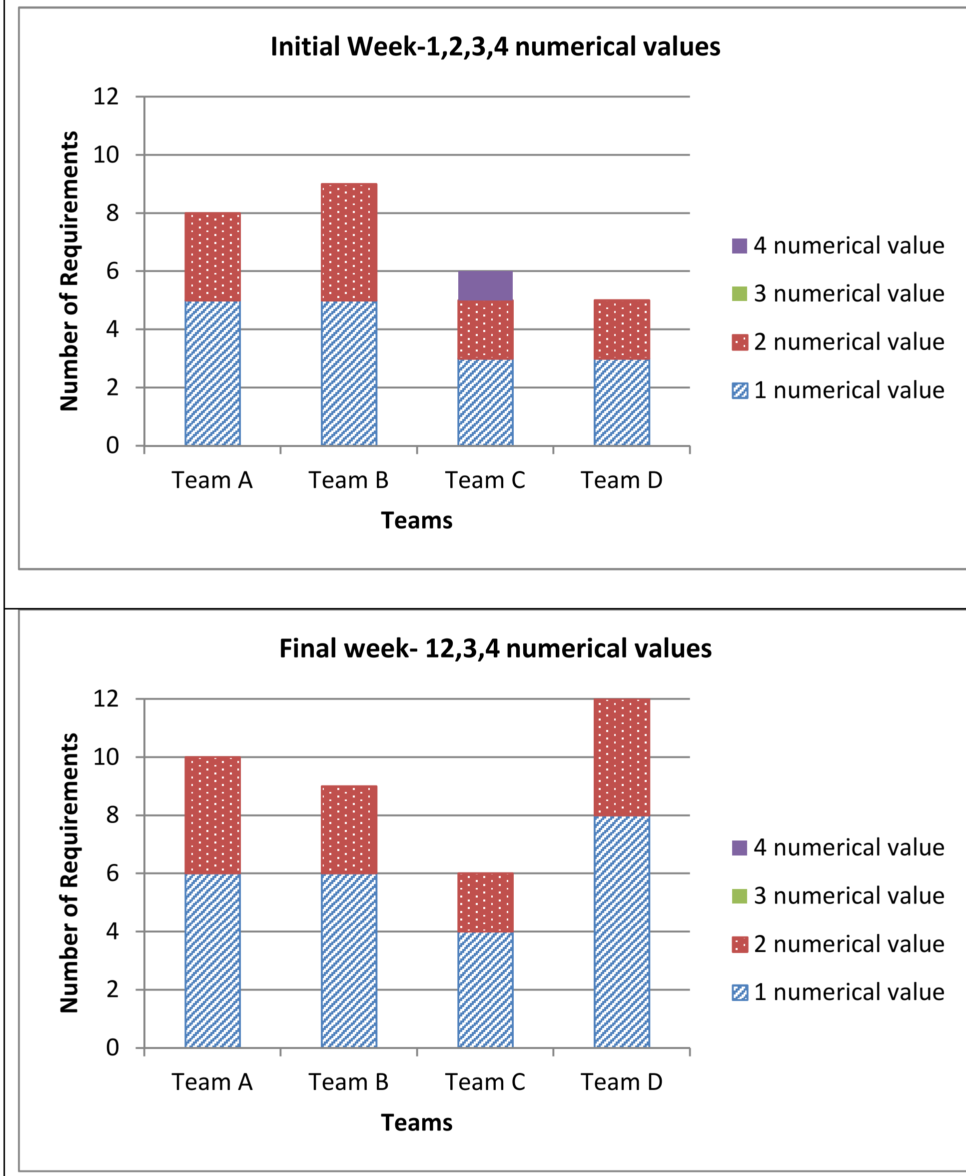


Figure 4-14 Comparing requirements with 1,2,3 or 4 numerical values for initial week and final week

It can be observed that across all four teams, there is increase in the number of requirements with at least one numerical value. This is expected because the requirements become more specific as they approach the completion of the project. There is also increase in the number of requirements with two numerical values for teams A and D. For team B, there is decrease in the number of requirements with two numerical values. For team C, though there is a decrease in the number of elicited requirements in final week (twenty in initial week to fourteen in final week), the number of requirements with two numerical values remains constant.

It is interesting to note that while most teams had requirements with three adjuncts in the final week, none of the teams have requirements with three numerical values either in initial week or final week. Further, only team C had a requirement with four numerical values in initial week. However, in the final week, this requirement was split into two requirements having two numerical values each. None of teams had any requirement with four numerical values in final week.

Thus, the findings from investigating the specificity in requirements as a count of numerical values suggest that the specificity in requirements increases from initial week to final week. This is reflected by increase in the number of requirements having one or more than one numerical values. This is similar to the observation made while considering the speficity of requirements as count of adjuncts.

Having numerical values within requirements is beneficial as the numerical values make the requirements measurable or testable. Further, numerical values within requirements are target values in most cases and thus it would be critical to meet these values for successful completion of the project. While the target values may not be established in the initial week, thus reflected through more requirements with no numerical values, it is desirable that the requirements have at least one target value towards the completion of the projects. The findings from this case study reveal that while the number of requirements with no numerical values decrease in final week compared to initial week, there are still some requirements with no numerical values. This may pose significant challenge while meeting these requirements because it would be hard to measure whether the requirements are met or not quantitatively. So for instance it is hard to measure whether the solution meets the requirement – *the design must be safe,* because there is not target value and the perception of "safe" can be qualitative. On the other hand, the requirement – *the design must withstand load of 100 lbs* is also a safety requirement but in this case whether or not the requirement is met is testable. Thus students should be encouraged to have at least one numerical value in the requirement as they approach the project completion.

Next, the specificity of requirements met is investigated.

4.9 Findings from Requirements met

RQ 2.3 aims at investigating whether or not more requirements with high specificity were met in the final solution as compared to requirements with low specificity. The requirements that were explicitly met in the final solution were counted and recorded. The specificity of the requirements met was then investigated.

Figure 4-15 illustrates the findings from investigating the specificity of requirements met while considering number of adjuncts as a measure of specificity. First looking at the requirements elicited, it can be observed that all four teams have elicited a high number of requirements with 0 and 1 adjunct. Across all four teams, the number of elicited requirements with 2 and 3 adjuncts decreases.

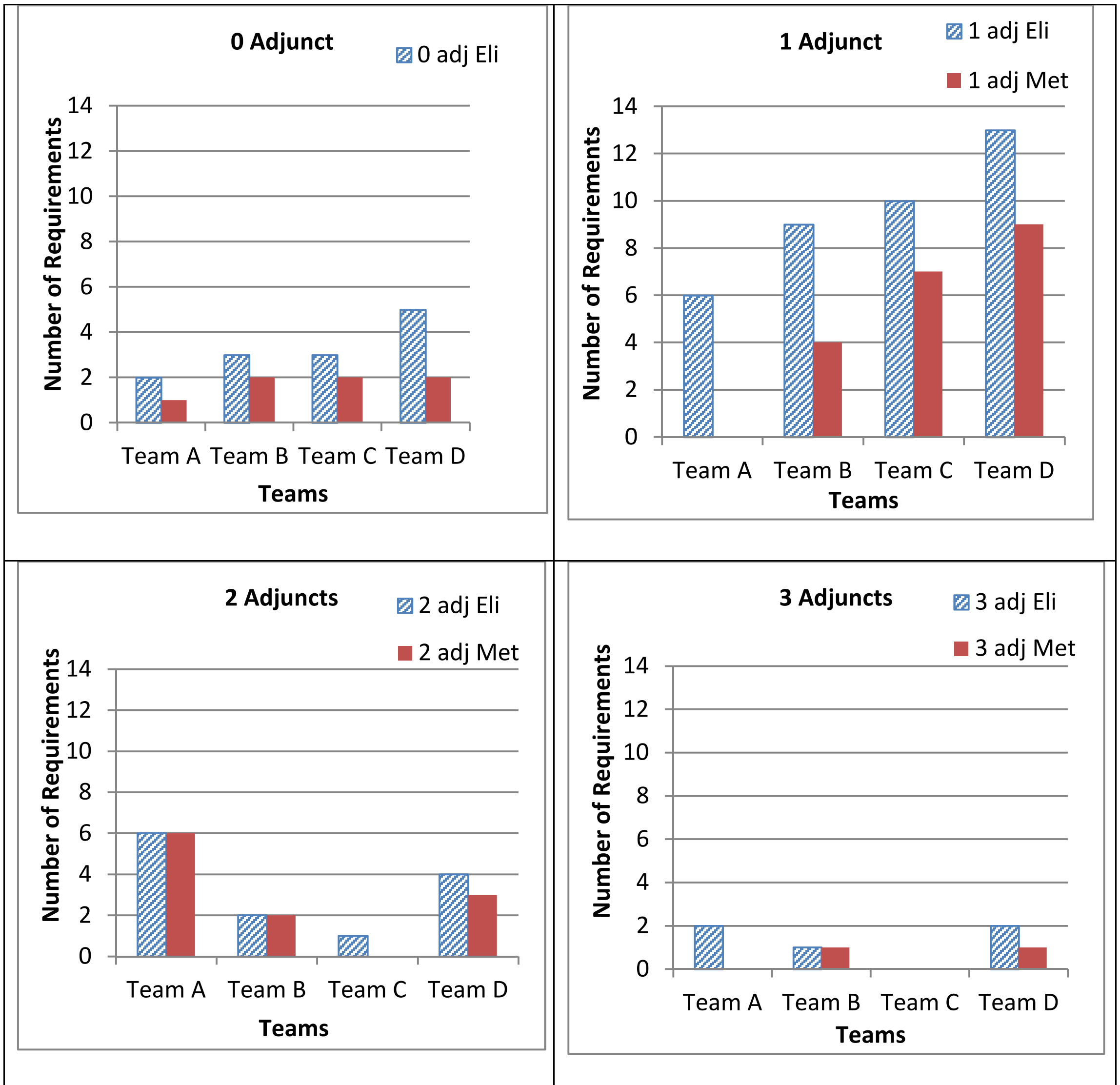


Figure 4-15 Comparing specificity of requirements met as count of adjuncts

Further, it can be observed that number of requirements met decreases as the number of adjuncts increase from 0 and 1 to 2 and 3. The only exception to this observation is team A that has most requirements met with 2 adjuncts as compared to requirements met with 0 and 1 adjunct. Similar observation can be made by comparing the average number of requirements elicited and met with 0, 1, 2 and 3 adjuncts across all four teams. As shown in Figure 4-16 the average number of requirements elicited and met is maximum for 1 adjunct. On average, the teams have elicited fewer requirements that are abstract (0 adjunct) or too specific (2 or 3 adjuncts). This pattern is also seen while meeting the requirements in the final solution. On average, the teams have met fewer requirements that are abstract (0 adjuncts) or too specific (2 or 3 adjuncts).

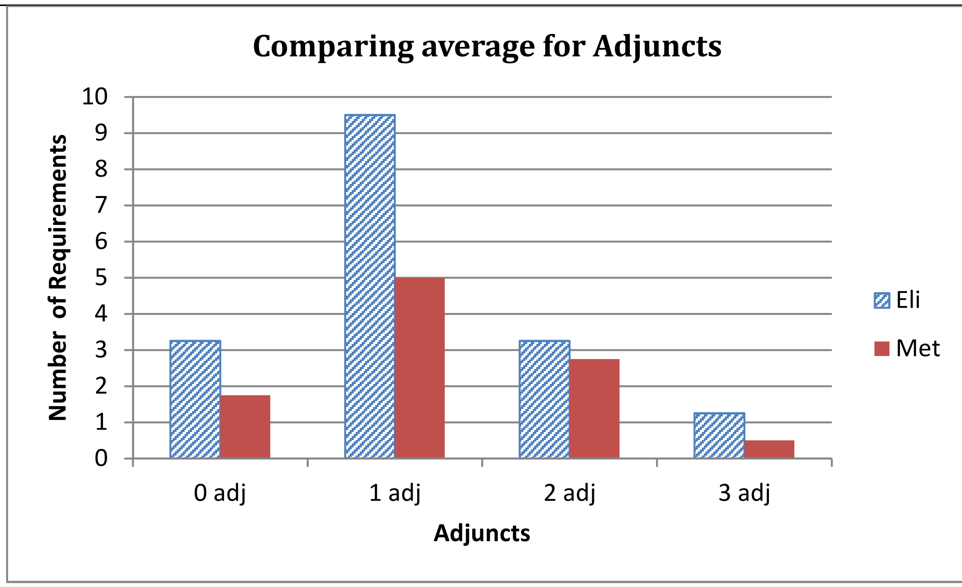


Figure 4-16 Comparing average requirements elicited and met for adjuncts

With increase in the number of adjuncts, the specificity of requirement increases. This also means that the constraint on the design space increases [17]. This makes it more difficult to meet these requirements. Thus while increase in number of adjuncts make the requirement more specific by increasing the level of detail, increase in adjuncts also makes it more difficult to address the requirement by constraining the design space. While there are no studies currently available suggesting an adequate number of adjuncts within a requirement, this study begins to show that too many adjuncts within the requirement might potentially jeopardize meeting those requirements. This study also shows that at novice designer level, requirements with 1 adjunct have the highest potential to be addressed in the final solution.

Figure 4-17 illustrates the findings from studying the specificity of requirements met while considering the number of numerical values as measure of specificity. It can be observed that all four teams have high number of elicited requirements with zero and one numerical values. All four teams have elicited fewer requirements with two numerical values. Further, it is evident from Figure 4-17 that teams have higher number of requirements met with zero or one numerical values. The number of requirements met decreases for requirements with two numerical values.

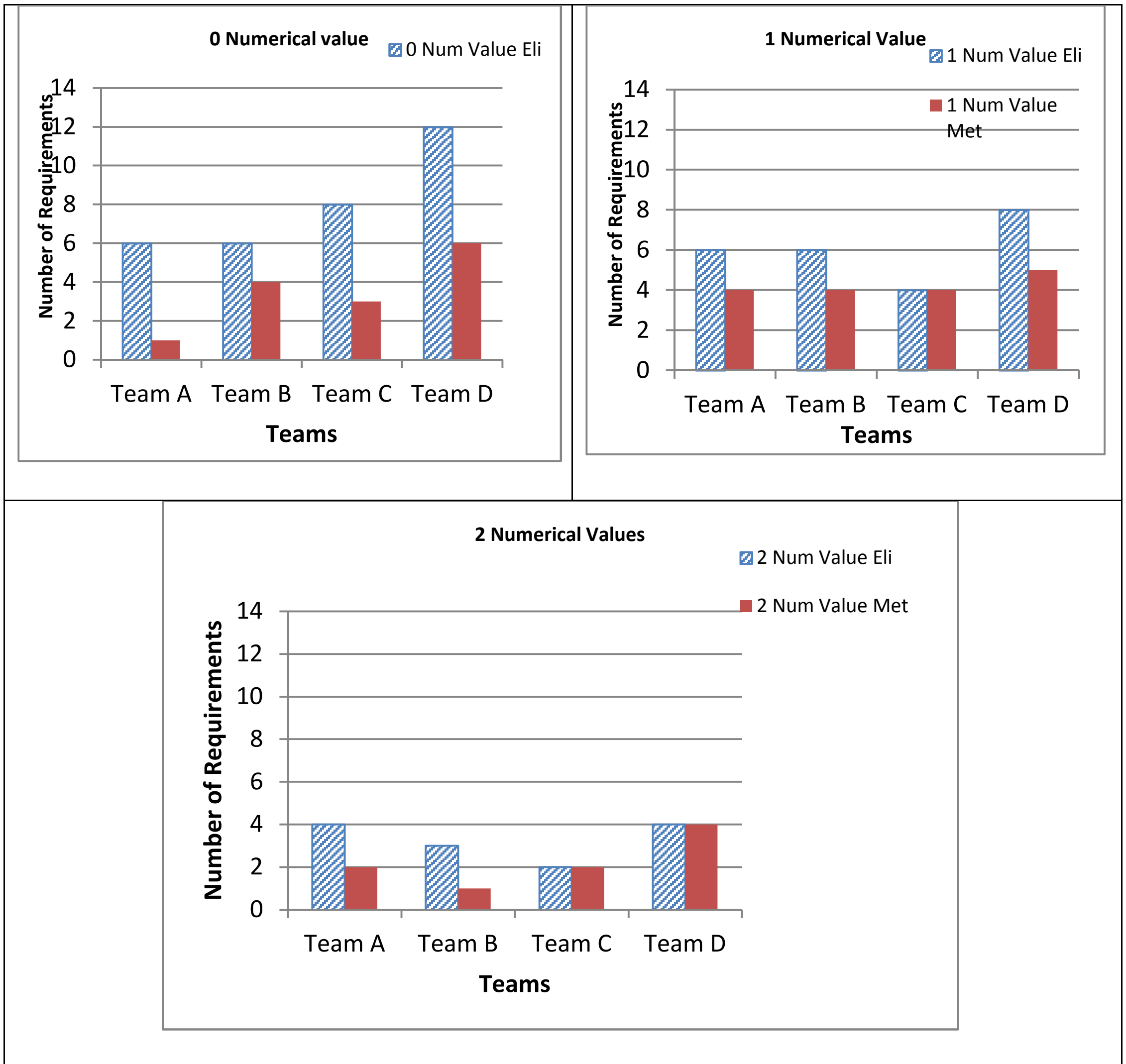


Figure 4-17 Comparing specificity of requirements met as count of numerical values

This is similar to the observation made for requirements with two or more adjuncts. Thus, as specificity of requirements measured as number of numerical values increases, the number of requirements met decreases. This observation can also be made while comparing the average number of requirements elicited and met for zero, one and two numerical values as shown in Figure 4-18.

On average, the teams have maximum requirements met with one numerical value. Further, on average, the teams have fewer requirement met with zero and two numerical values.

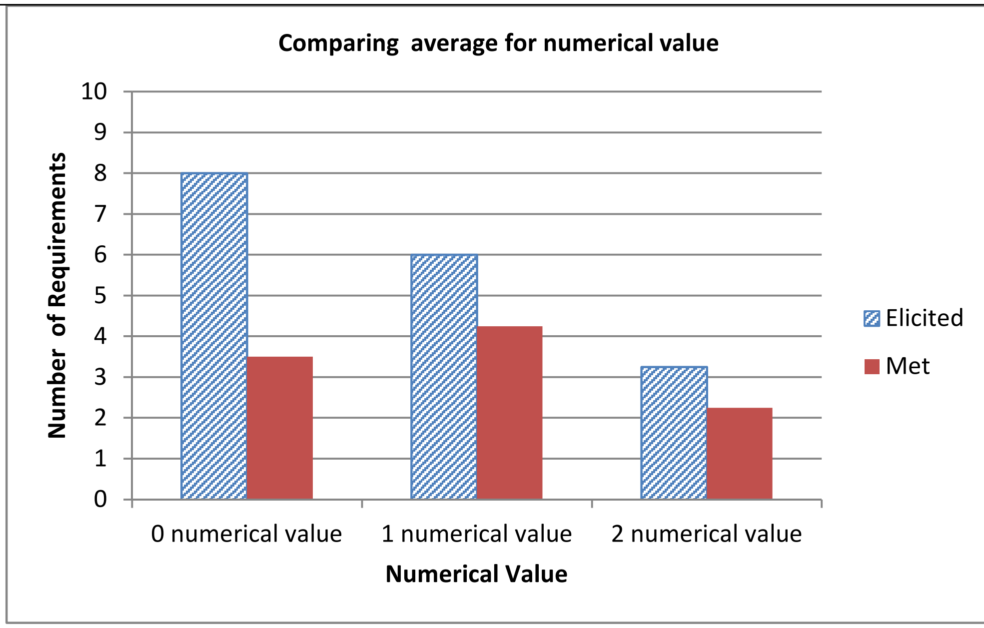


Figure 4-18 Comparing average requirements elicited and met for numerical value

Studying the specificity of requirements met as count of adjuncts and numerical values suggest that the requirements with one adjunct or one numerical value have higher potential of being met in the final solution when compared to requirements with either less than one adjunct or numerical value or more than one adjunct or numerical value. A project is considered successfully complete when all the established requirements are met by the final solution. Since the requirements with either one adjunct or one numerical value have higher potential of being met, the students should be encouraged to elicit requirements with at least one adjunct or numerical value. Requirements that have no adjuncts or numerical values and the requirements that have too many adjuncts or numerical values are potential red flags and may not be met.

4.10 Findings from Requirements tracing study

RQ2.4 aims at investigating the evolution of requirements within student teams and identifying specific changes within each requirement. As previously mentioned, three aspects are considered here 1)number of addition, deletions, changes and no changes, 2)identifying changes in natural language requirement elements (system, necessity, behavior, object and condition) and 3)identifying change types as per the change taxonomy established in [16, 65].

First the findings from studying the number of additions, deletions, change and no change are discussed in Section 4.10.1. Section 0 then discusses the findings from changes in the natural language requirement elements. Finally, Section 0. discusses the findings from change types based on change taxonomy.

4.10.1 Number of additions, deletions, change and no change

The first evolution type investigated in this case study is the high level modification of the requirement document. This includes changes such as addition of a new requirement, deletion of existing requirement, changing an existing requirement or keeping a requirement as is. The findings are illustrated in Figure 4-19.

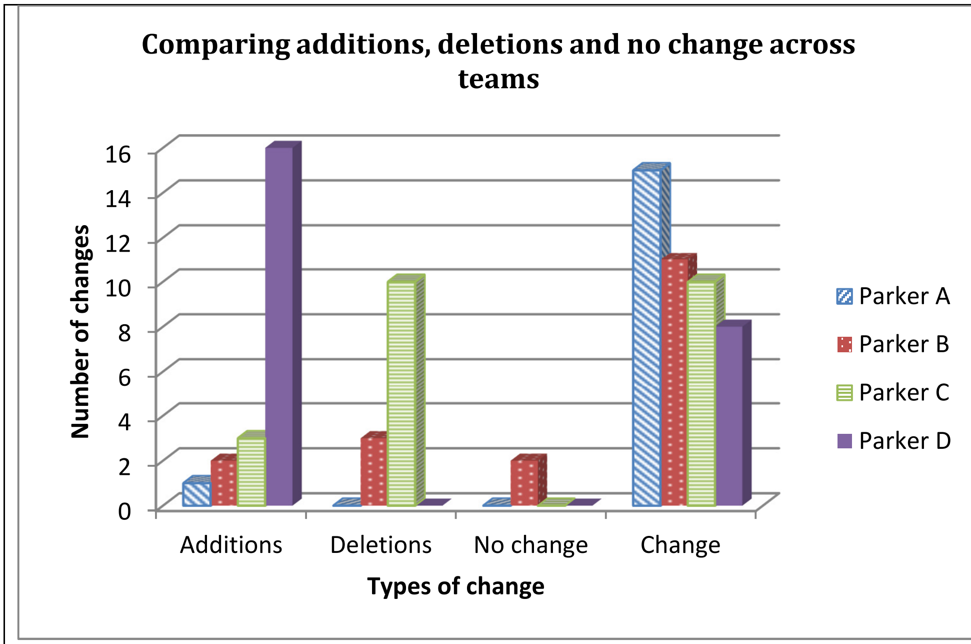


Figure 4-19 Comparing additions, deletions, change and no change across teams

From Figure 4-19, it is clear that all teams have multiple types of changes in the requirement document. Additionally, all four teams had a very high number of 'change' indicating that the requirements changed significantly from initial weeks to final week. Team D had eight requirements in initial week and twenty four requirements in final week resulting to the most number of additions. Team C, on the other hand, had twenty requirements in initial week and fourteen requirements in final week resulting to the most number of deletions. Of the four teams, only team B has requirements with "no change".

These findings suggest that requirement document of novice designers' change significantly from initial weeks to final weeks. Further, these changes are not just limited to addition of new requirements as the understanding of the problem grows but also include deletions and modifications of existing requirements. From the data currently available, the novice designers have not stated justification for additions, deletion or modifications to the requirements. It is interesting to note that other documents such as weekly summary, presentations, mid-term and final reports that the students are required to submit as a part of project deliverable do not mandate providing updates on requirements. None of these documents require the students to provide justification for the changes in requirements document. Based on the findings of this case study, it is known that the requirement document changes significantly. If there is no justification for these changes, valuable information could be lost. Further, the novice designers will not be able to trace if a requirement was accidently deleted or added if they do have the documented justification for additions and deletions of requirements.

After investigating the document level changes, the next step was to investigate changes in the natural language requirement elements. This is discussed in 4.10.2.

4.10.2 Changes in natural language requirement elements

The second evolution type investigated in this case study is tracing the change in natural language requirement elements [17] . This accounts for changes in the system, necessity, behavior/characteristics, object or condition of a requirement. Figure 4-20 illustrates the overview of the changes in the natural language requirements elements.

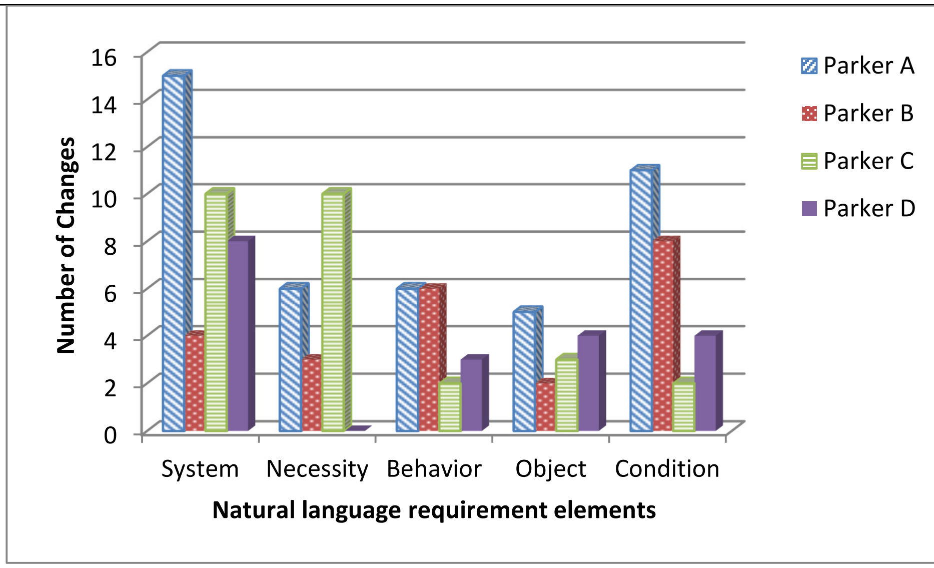


Figure 4-20 Overview of changes in natural language requirement elements

From Figure 4-20, it can be observed that all four teams had multiple changes in the natural language requirement elements. Most number of changes was observed in the 'system' element. The least number of changes were found in the object element of the requirement sentences. The specific changes in the natural language elements of requirement sentence are discussed in the following paragraphs.

As previously mentioned, most number of changes was observed in the system element. Figure 4-21 illustrates the summary of system changes for all four teams.

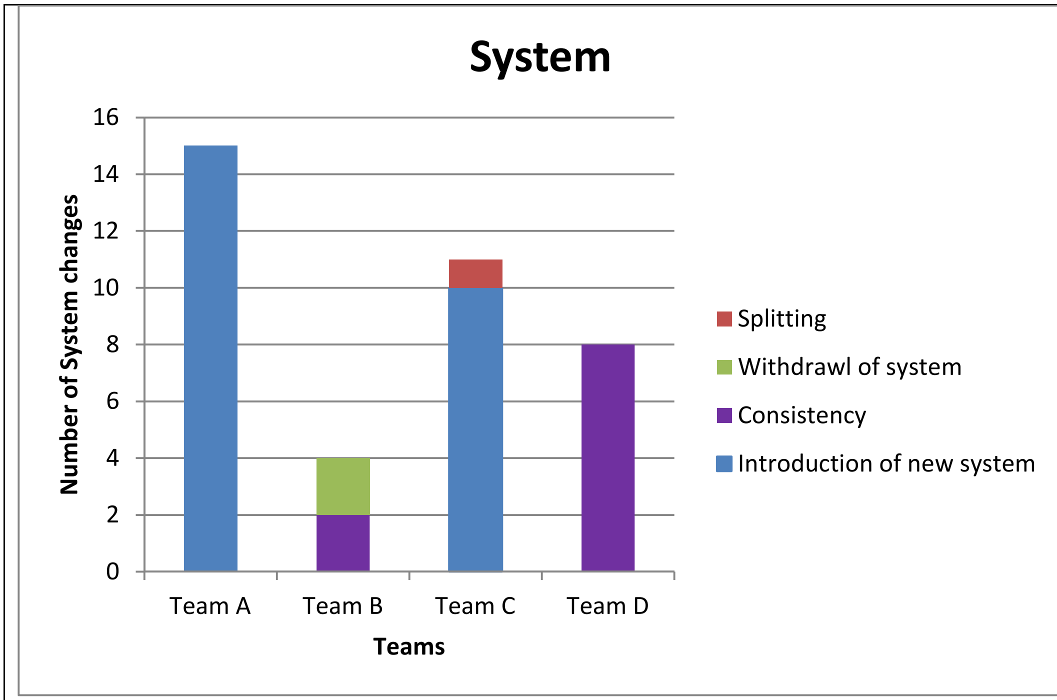


Figure 4-21 Summary of system changes

Syntactically, system is represented by the subject within a sentence. Thus, the system changes also represent the change in the subject of the requirement sentence. As observed in Figure 4-21, of all the four teams, team A had maximum system changes (sixteen) while team B had the minimum system changes. Further, there were several types of system changes observed across the four teams. Studying the requirements tracking sheet for team A, it was found that the change in system essentially stemmed from addition of a system (subject) in the requirements within final week requirement document. Similar observation for made for team C where the system changes stemmed for addition of a system (subject) in the final requirement document. For team D, however, the system changes stemmed from changing the subject from 'system' in initial week document to 'final design solution' in the final week document. Team B was the only team that showed removal of the system from one of the requirements.

Next, Figure 4-22 illustrates the necessity changes across all four teams. The element of necessity within a requirement statement is syntactically represented by a modal and it shows the importance of the requirement [17].

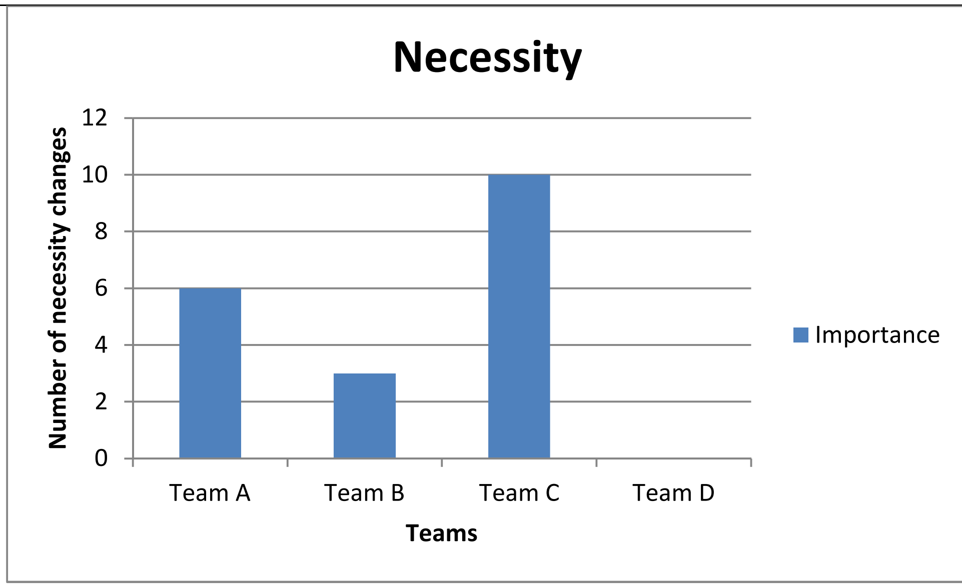


Figure 4-22 Summary of necessity changes

Necessity element of requirement changed for all teams except team D. Of the remaining teams, team C had the highest change in the necessity while team B had the lowest change in the necessity. Investigating the requirements tracking sheet for all four teams, it was found that for most teams, the change in necessity stemmed from addition of a modal such as must or should in the final week requirements document.

Next, the element of behavior within a requirement represents the function of the system or component being designed and is syntactically represented by a verb [17]. Figure 4-23 illustrates the summary of behavior changes across all four team.

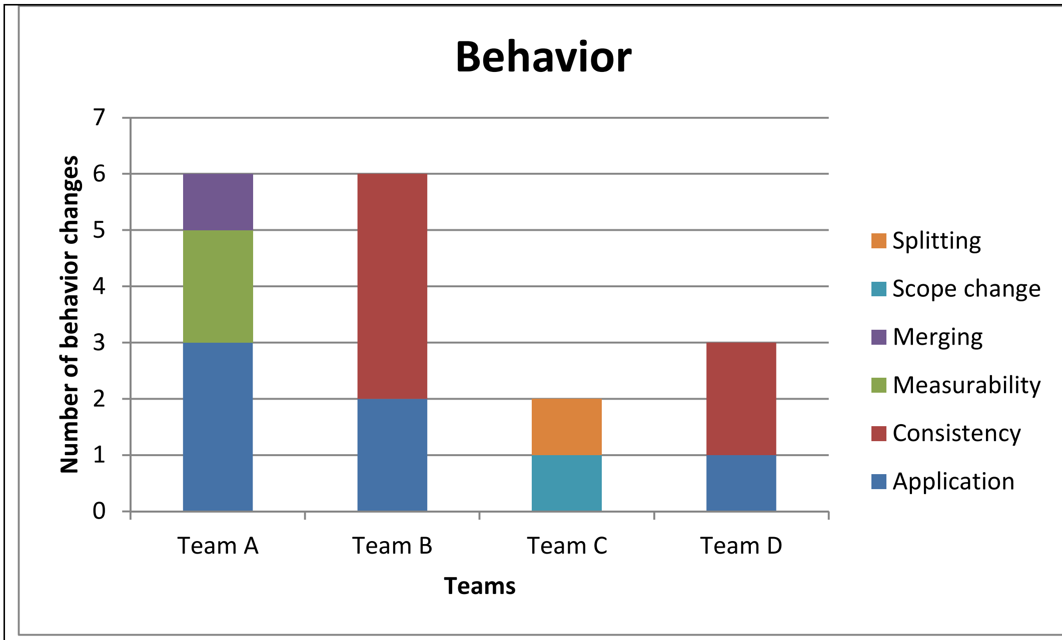


Figure 4-23 Summary of behavior changes

From Figure 4-23, it can be observed that Teams A and B had maximum changes in the behavior element while team C had minimum number of behavior changes. It is evident that there were varieties of behavior changes within the requirements for all teams. Upon further investigating the requirements tracking document, it is found that most changes in the behavior stem from either change in the function of the system (for example –must "remove" excess adhesive and must "minimize" excess adhesive) or inconsistency in the vocabulary used to describe the function of the system (for example – must "provide" economic advantage and must "present" economic advantage). These can be seen in Figure 4-23 as high number of application and consistency changes respectively. Apart from these two, the behavior changes also resulted from merging of two requirements, splitting of a requirement into multiple requirements and change in the scope or measurability of requirement.

The next element is object and it represents what the system is affecting [17]. The summary of object changes is illustrated in Figure 4-24.

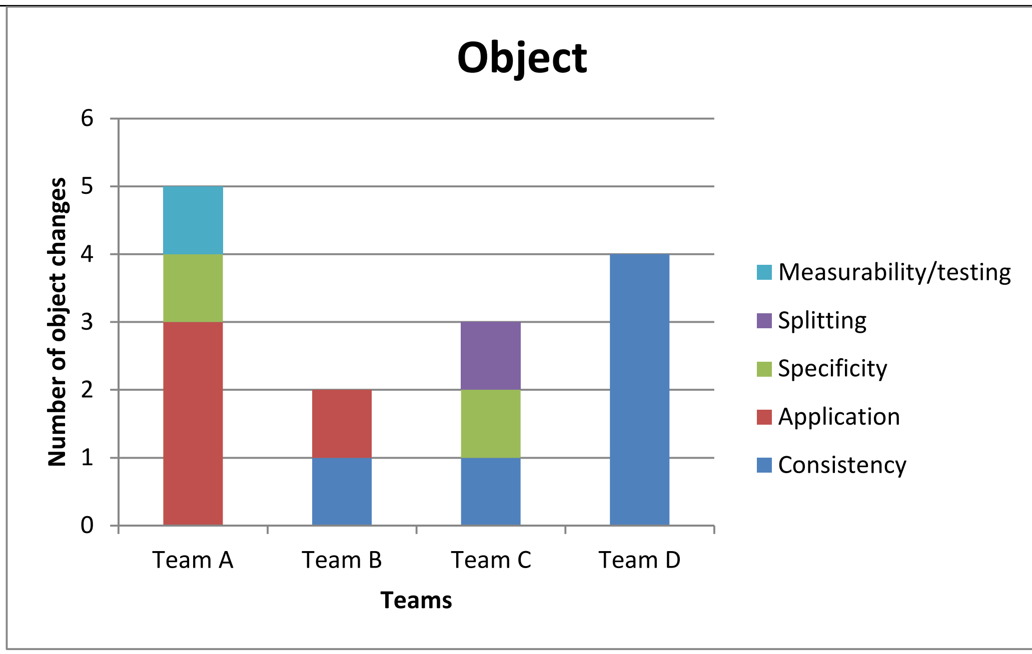


Figure 4-24 Summary of object changes

From Figure 4-24, it can be observed that all four teams had changes in the object element with team A having the most and team B having the least number of object changes. Similar to the changes in other elements, several type of changes were observed in object element. Investigating the requirements tracking sheet, it was found that most changes in the object of a requirement stemmed from change in the vocabulary within requirements. So for instance, one of the requirement for team D changed from "*the system must count completed spliced parts*" to "*The final design solution must count spliced O-rings*". Here the object changed from "spliced parts" to "spliced O-rings", which is essentially inconsistency in the vocabulary as the completed spliced parts are the O-rings.

Another change in the object of a requirement stemmed from the change in the application and this was observed in the requirements of teams A and B. So for instance, one of the requirements of team A changed from "*Must handle **extrusions** ranging in diameters of 0.070" to 0.250*" to "*The design must form* **o-rings** *from extrusions ranging in cross-sectional diameter from 0.070 inches to 0.250 inches"*. Here the object changed from "extrusions" to "O-rings" as a result of change in the application of system from handling extrusion to forming O-rings. Other changes in object of a requirement resulted from splitting of requirement, change in measurability of specificity of requirement.

After object, the next natural language element of a requirement is condition. Of the five natural language elements, the second highest change was observed in the 'condition' element as can be observed in Figure 4-20 which shows the overview of all elemental changes. The summary of condition changes is illustrated in Figure 4-25. The condition of a requirement changed in multiple ways for all teams. Further, it can be observed that condition element of requirement has most different types of changes.

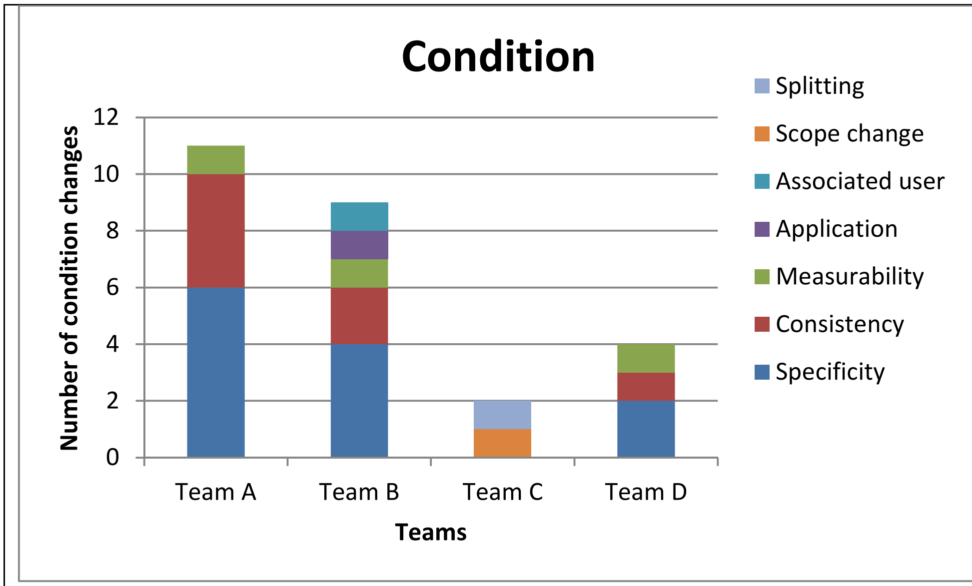


Figure 4-25 Summary of condition changes

Team A had the maximum while team C had minimum number of condition changes. Changes in the condition of a requirement are essentially changes in the adjuncts or complements. These describe the level of detail within a requirement [17]. Upon investigating the requirements tracking document for each team, it was found that most changes in the condition stemmed from change in the vocabulary or change in the specific details of the requirement. For example, condition for requirement-2 for team B changed from "ranging from 2 inch to 2 feet in ring diameter" in initial week to " ranging from 2 inch to 24 inch in ring diameter" in final week. Here the unit changed from 2 feet to 24 inches. Other type of condition changes resulted from change in measurability, application, change in associated user, change of scope or splitting of a requirement. These change types are further discussed in Section 4.10.3.

This section describes the changes in the natural language elements of a requirement. Within a single requirement, the element 'system' represents the artifact being designed [17]. It is therefore a critical element of the requirement because without a 'system', the designer would not know what the requirement is for. The element 'necessity' describes the importance of a requirement and would be critical to distinguish between a constraint and criteria [17]. Behavior indicates what the system being designed must do and condition adds to the level of detail of a requirement [17]. Each of these element is critical within a requirement and any changes must be documented with the justification for the change [16].

From the case study findings discussed in this section, it is evident that the all the elements of a requirement change from initial week to final week and the change occurs in multiple different ways. Further in most cases, the change is observed across all four teams. Without documented justification for each of these changes, the information about why a particular change was made in a requirement is lost. Thus, the novice designer cannot track down if a change was accidently introduced in a requirement. A survey of design textbooks was conducted to investigate the tools pertaining to requirements [72] and none of the textbooks describe tools for managing the requirement changes.

4.10.3 Change types as per change taxonomy

The next evolution type investigated within the requirement documents was to trace the 'type' of changes as per the change taxonomy established in [16]. It may be noted that while the taxonomy captured all the change types occurring in student requirement document, not all the change types described in the taxonomy were found in the requirement documents. Example of some of these change types include replacement of system, updating, incorrect raw data interpretation and correcting among others. Figure 4-26 illustrates the summary of change types for all four teams.

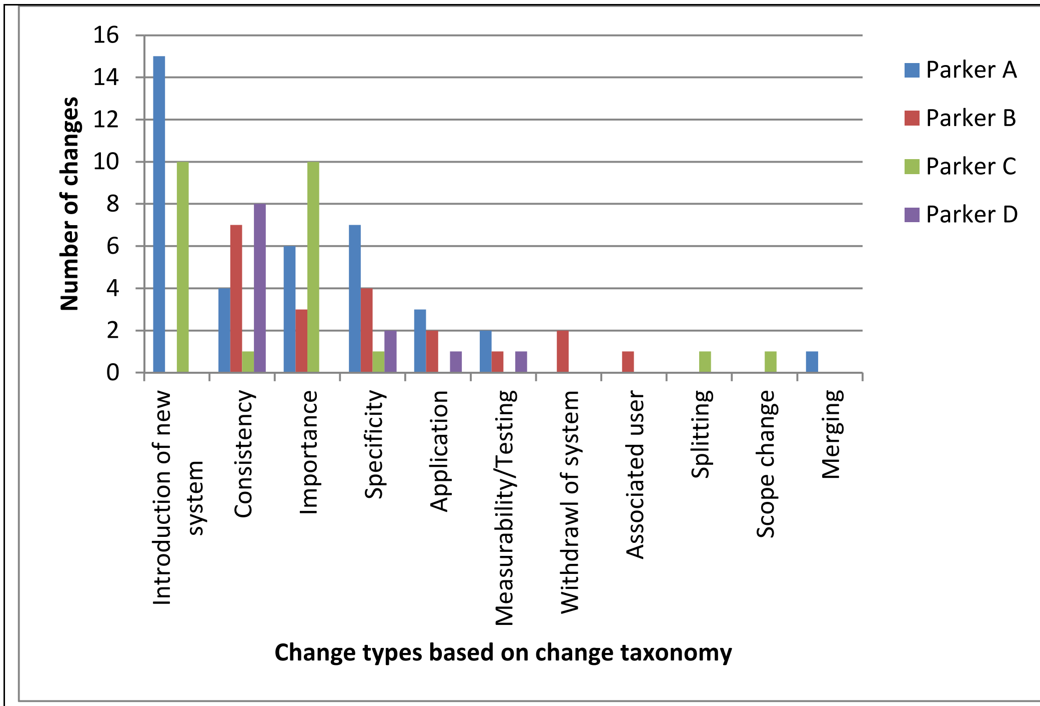


Figure 4-26 Summary of change types based on change taxonomy [16]

Again, it can be observed from Figure 4-26 that all teams had multiple types of changes in the requirements. With the introduction of a new system being the most frequent change type. This change was observed in teams A and C as a result of addition of the subject in the requirements for final week. Other change types that are more frequently observed across all four teams include consistency, importance and specificity. These changes stemmed from change in vocabulary or units, addition or deletion of a modal and change in the specific details of requirement respectively.

The change types that are less frequent include application, measurability/testing and withdrawal of system, while the change types associated user, splitting, scope change and merging only occurred once. Further, the changes that are less frequent were mostly accompanied by other change types that are more frequent. So for instance, splitting and merging occurred once, but in each case they were accompanied by introduction of a new system.

Some of these changes can be more critical than others and if not properly document can lead to information loss. For instance the change types such as splitting or merging involve either separating a compound requirement or uniting two requirements into one. This may not always lead to major change in information as it is essentially re-writing requirements in different form. For example the requirements "(*the system) Must handle organic and silicone adhesives*" and "(*the system must) apply adhesive to ends of extrusion*" were merged into one requirement "*The design must have the capability to apply either organic or silicone adhesive to an extrusion end*" which is essentially conveying the same information.

On the other hand the change types such as introduction of new system, importance, specificity and consistency which can alter the purpose of a requirement can prove to be more critical making the documentation of these changes more important. For example for team C, the requirement in initial document changed from "*Apply **controlled amount of** organic or silicone adhesive to extrusion ends*" to "*The system must **apply 1 drop per square inch** of organic or silicone adhesive to extrusion ends"* in the final document. Here the team added specific detail about the "controlled amount" and thus this change is specificity change. Adding information about how much adhesive enhanced the purpose of this requirement making this change more critical compared to just merging or splitting a requirement. It is interesting to note that the changes which are more critical are also more frequently observed in requirement documents of the teams. Again, if the students are not taught to identify these change types, the criticalities associated with the change types and appropriate tools or methods to document and track these changes; it can lead to potential loss of valuable information and jeopardize the successful completion of the projects.

4.11 Summary of RQ2 – How are Students Using Requirements?

RQ2 aims at investigating how are students using requirements. To that end, four different aspects are investigated: 1) mapping the delta in requirements to level of detail of final solution, 2) completeness and specificity of elicited requirements, 3) completeness and specificity of requirements met, and 4) tracing requirements to identify change types.

Investigating the delta in requirements between student and sponsor requirements reveals that though working on the same project, students have different understanding of the design problem. While it is expected that the students have a difference in the understanding of the problem towards final week since all the teams are working independently, it is expected that they have similar understanding of the problem in the initial weeks. Considering only the number of requirements, three out of four teams had more requirements than the sponsor in the initial week. Team D, however failed to identify the requirements established by the sponsor. This shows a lack of uniformity and systematic methodology of eliciting and documenting requirements between teams.

Further, mapping the delta in requirements to the level of detail in the final solution, it was found that the team with consistent delta had a high level of detail in the solution. The teams with changes in the delta from initial week to final week had medium level of detail in the final solution. This shows that inconsistencies in understanding the problem or realizing the project requirements much later in the process could jeopardize the level of detail in the solutions.

Next, the completeness and specificity of requirements in initial week and final week are investigated. While completeness is measured considering linguistic components of requirements such as subject, modal and verb phrase, the specificity is measured as a count of adjuncts and numerical values within requirements [17]. The findings from completeness study suggest that the completeness of requirements increases from initial weeks to final week. This is expected because the understanding of the problem increases as the students' progress in the design project.

Similar observation is made while investigating the specificity in requirements. From initial weeks to final weeks, the specificity in requirements increases as indicated by increase in number of requirements with more adjuncts and numerical values. Again, this is expected as with the progress in the design process the requirements become more specific. It can also be inferred from this that as the requirements become more specific, the designers are approaching the end of the project. One the other hand, if the students are approaching the end of project according to the timeline, but have poor specificity in the requirements, it could potentially mean that the students may not complete the design project successfully. This then, leads to the recommendation that the students must be taught to use completeness and specificity as measure of approaching project completion internally within their teams.

Next, the specificity of requirements met in the final solution was investigated. The findings reveal that the requirements that had low adjunct or low numerical value were met more frequently compared to more specific requirements (having three or more adjuncts or 2 or more numerical values). While it is desired that the level of detail in the requirements increases towards the end of the project, more specificity in the requirements may constraint the design space too much making it challenging to meet those requirements in the final solution.

Finally, the changes within the requirement document of novice designer were investigating. The findings suggest that the requirements document of novice designer changes in multiple different ways. The change occurs at document level in terms of additions and deletions of requirements and at micro level in terms of change in the specific elements of requirements. In current practice, the novice designers are not taught tools that would help them manage these changes. This could lead to loss of potential information about project requirements.

The takeaways from the case study to investigate how students are using requirements are summarized below:

- Currently, the students do not seem to have knowledge about formal methods of eliciting and documenting requirements. Teaching the students appropriate tools and methods to formally elicit requirements will increase the likelihood of project success as this will be reflected in the increased level of detail of final solution.

- Completeness and specificity increase towards the completion of the project. Students can be introduced to the completeness and specificity aspects of requirements and can be taught to use these as an internal measure of project completion. Further, increasing the specificity of requirements by adding appropriate numerical values also make the requirement measurable or testable.
- While increase in specificity is desired to have appropriate level of detail in a requirement, a requirement that is too specific can constrain the design space unnecessarily. Students can be taught to identify requirements with too many adjuncts or numerical values as requirements that may potentially not be met.
- Requirements document of novice designers change from initial week to final week and the change occurs in multiple different ways in multiple different requirement elements. Currently, there is no system in place that would allow the novice designers to track or manage these changes. No tools are found in design textbooks that educate the novice designers about managing requirement changes. It is strongly recommended that students are educated about the requirement changes and potential dangers of not managing them well. Perhaps the requirement update sheet used to collect data for this case study can serve as starting point in this direction.

While, only one capstone project was investigated in this case study, the findings can be extended to other projects because the students are taught follow a similar design process and practices irrespective of the design problem given to them. However, there are several limitations associated with this case study and these are explained in section 4.12.

4.12 <u>Limitations of the Case Study</u>

The case study was conducted with single capstone design project consisting of four teams working simultaneously to solve the design problem at hand. As previously mentioned, while the findings can be extended to other capstone design projects, there are several limitations of this case study.

Several case studies are conducted using the information generated by students [41, 43]. However, few of these studies require the subjects under study to generate the data that they would otherwise not generate as a part of natural process. For the purpose of this case study, the elements under study –the students, were asked to collect the data in the form of requirements update sheet. While the students are asked to generate documents such as weekly summary, power-point presentation and mid-term and final reports as a part of course requirement, they are not required to generate weekly requirements update document. Thus, asking the students to collect the data by submitting the weekly requirements update sheet may alter the natural project execution behavior of the students leading to potential bias in the study.

Historical case studies have been conducted by several researchers [39, 73, 74] where they investigate the information already generated in completed projects to answer the research questions. Studying historical data allows mitigating the bias in the study because the researcher is not required to interfere in the natural execution process.

Conducting a historical study to investigate how students use requirements would be challenging as none of the documents currently generated by the students capture the weekly requirement updates required for this study.

One way to address this is to conduct the case study by having the researcher act as a passive participant in the project and thus collect the data through observations [43, 42]. Having the researcher as passive observer could be beneficial as it would provide contextual information however; the data collected could potentially be subjective [75] and thus biased. Again, there is a possibility of altering the natural behavior by adding an external observer in the project.

The findings from this case study reveal the potential benefits of using a tool or method to track the requirement changes. While not currently required, the students can be mandated to submit the requirement update sheet as a weekly deliverable for the project in addition to the executive summary and presentations. This would be similar to the process followed in several industries.

Thus, using the requirements update sheet will have several benefits. First it will allow the students to manage and track requirement changes. Second, it will train the students for the practices followed in industry. Finally, it will allow future researchers to collect the data to study requirements without intruding in the natural process.

CHAPTER FIVE: WHAT IMPACT DO REQUIREMENTS HAVE ON OUTCOMES- A DESIGNER STUDY

5.1 Study Objective and Overview

The goal of conducting the designer study is to explore and understand the influence of requirements elicitation activity on idea generation. Specifically, the focus is on the type and number of requirements elicited and number of requirements addressed in the design solution. Currently, it is not clear to what extent the designer involvement in requirement elicitation and definition affects idea generation. It is important to develop this understanding, as this knowledge can then be used to develop systematic guidelines for use of requirements in idea generation activities and tools. Further, if a positive correlation is observed between the designers' involvement in requirements elicitation and the idea generation activity, it could lead to the need for integrated design team activities in the problem definition stage. Thus, this leads to research question – *What impact do requirements have on outcomes?* Table 5-1 illustrates the summary of sub-questions and hypothesis.

Table 5-1 Summary of sub-questions and hypothesis for RQ3

Research Question	Hypothesis
RQ3.1 When assigned the task of eliciting requirements, what types of requirements are elicited by novice designers?	H3.1 When assigned the task of eliciting requirements, mostly functional requirements are elicited by novice designers
RQ3.2 Does the involvement of designer in the task of elicitation impact the number of requirements addressed by them?	H3.2 Involvement of designer in the task of elicitation will lead to more number of requirements addressed by them.
RQ3.3 Does having a list of requirements cause fixation when addressing more requirements?	H3.3 Having a list of requirements will cause fixation when addressing more requirements.

To answer this research question, a designer study was conducted with three groups: 1) 0(yes)-Group - Requirements elicitation group, 2) 5(yes)-Group - Partial requirements elicitation group, and 3) 10(no)-Group - Control group or no elicitation group. Table 5-2 provides an overview of the experimental conditions for each group.

As can be seen in Table 5-2, the 0(yes)-Group will only be given the problem statement and will be tasked with eliciting the requirements and developing concepts. The 5(yes)-Group will be given the problem statement and some requirements. Using this information, they will be tasked to elicit more requirements and generate concepts. Finally, the 10(no)-group will be provided with the problem statement and the full set of requirements. Their only task will consist of generating the design concepts.

Table 5-2 Summary of experimental Groups

Groups	0(yes)-Group Requirements elicitation group	5(yes)-Group - Partial requirements elicitation group	10(no) Group - Control group.
Condition	Given only problem statement	Given problem statement and some requirements.	Given problem statement and list of requirements
Procedure Brief	• Given 5 minute to read and understand instruction and ask questions • Given 10 minutes understand problem and elicit requirements • Generate ideas – 20 minutes	• Given 5 minute to read and understand instruction and ask questions • Given 10 minutes understand problem and elicit requirements • Generate ideas - 20 minutes	• Given 5 minute to read and understand instruction and ask questions • Give 5 minutes to understand problem and requirements • Generate ideas -20 minutes

The remainder of this chapter provides details associated with the participants, the design problem, and the experimental execution procedure for each group.

5.2 Participants

The participants for this study consisted of senior level mechanical engineering students from ME-402 class. These students are selected for this study as they would have acquired the necessary formal design education in terms of their familiarity with process of requirements elicitation and idea generation.

A total of forty five students participated in the study with fourteen students in 10(no) group, fifteen students in 0(yes) group and sixteen students in (5(yes) group. The participants performed the design task individually. While the case study is focused on understanding the requirements elicitation in design teams, the focus of the designer study is primarily to understand requirements elicitation primarily by individual designers, thus the students performed the design task individually.

The students were not awarded any extra credits or reward for their participation. Awarding extra credit or rewards for participation can lead to possible bias in the quality of solutions generated by the students [76]. This can skew the experiment results and thus in order to avoid this bias no extra credit will be awarded.

The experiment was performed in the regular class setting. The environment in which the experiment is conducted can have a significant effect on the results [76]. The environment that is familiar to the population, such as the classroom, lab, or office may be more conducive than unfamiliar environment. The factors such as ambient noise, light, surroundings can also influence the experiment. While, external factors such as ambient noise are out of control, conducting the experiment in familiar environment such as the classroom can help to reduce the bias and therefore the experiment was conducted in the regular class setting.

Further, the students were randomly selected making sure that there is roughly the same number of students for testing each of the three conditions [76, 77, 78]. It may be noted here factors such as gender, individual skills, knowledge and experience can cause bias in the results; however investigating the bias due those factors is out of scope for this experiment.

5.3 Designing a Problem

Selecting an appropriate design problem is a critical aspect of conducting a designer study because factors associated with design problem such as scope and description of the problem can significantly influence the experimental results [76].

A common design problem was given to all the participants for the study. Figure 5-1 illustrates the peach picking design problem assigned to all the participants for the study.

People in wheel chair have a very limited range of reach especially for heights. With this limitation, they cannot experience pleasures such as that of picking peaches in summer. Design a device that will allow users in wheel chair to experience the joy of picking peaches from the tree and collect it in a basket while still in the wheel chair. The device must be manually operated and prevent damage to the fruit while picking it. The fruit should not fall on the ground while picking.

Figure 5-1 Peach picking design problem

In addition to the design problem, the participants of 5(yes) and 10(no) group were given a set of requirements, these can be found in Table 5-3.

Table 5-3 Requirements given to NE and PR groups

Requirements	NE-group	PR-group
The device must reach heights in the range of 8 to 10 feet while used by a person in wheel chair.	X	X
The device must allow the user to pick multiple fruits at a time.	X	X
The device must grasp the fruit.	X	
The device must hold the fruit until the fruit is put in the basket.	X	
The device must provide an indication to the user when the fruit has been picked.	X	
The forces required to operate the device must be within the upper body strength of a person in wheelchair.	X	X
The device must be safe to use.	X	
When not in use, the device should fit in 4 feet by 3 feet storage space.	X	X
The device must not be made of corrosive materials.	X	
The device must not cost more than $ 50.	X	X

The peach picking problem was selected by considering few criteria which are discussed below:

1. The problem should be within the knowledge domain of senior level mechanical engineering students; however it should not be a familiar problem. Familiarity with the design problem may result to bias when generating design concepts [79, 80] .While familiarity of the design problem will not be explicitly measured, it will be ensured that solution to the problem currently does not exist in real world [80]. Some examples of such problems are designing an automatic clothes ironing machine [79] or automatic burrito folding machine [80]. While most students are familiar with the process of ironing clothes or folding a burrito, there is no solution currently available in the market for these problems. Thus, although the problems fall within the knowledge domain, the students are not familiar with the

solution to the problem. Similarly, while most students are familiar with the process picking peaches from the trees, there is no solution currently available in the market for this problem. Thus, although the problem falls within the knowledge domain, the students are not familiar with the solution to the problem.

2. The problem should be appropriately represented to avoid any gender bias. In order to fulfill this, the gender of the customer was not explicitly mentioned in the description of the design problem. It was left to the interpretation of the designer. Further, both male and female participants would be familiar with and may have experienced the process of picking a fruit from a tree. Thus, precautions were taken to avoid gender bias through the representation of the problem.
3. The problem should be based on the real need as this will motivate the students to solve it [78]. This will be achieved by describing a customer story and establishing a need for designing solution through the problem statement description. This was achieved by describing the customer, the limitations at the customer end and establishing a need for designing solution through the problem statement description.
4. It is necessary that the students are able to solve the problem in the stipulated time. If the problem is too complicated to solve in the given time, it may lead to insufficient data for the experiment. A pilot study was conducted to ensure that the students will have sufficient time to solve the problem.
5. It is essential that the problem allows for a possibility of generating many concepts. This will aid in gathering sufficient data from the study. Again, a pilot

study was conducted to ensure that the problem allows for generating at least more than 5 concepts.

6. Since one of the goals is to study the requirements elicited by students, the design problem should allow for elicitation of requirements. In order to verify that the given problem allows for eliciting requirements, a pilot study was conducted with members of CEDAR lab. For this study, the participants were provided with sample design problems and the peach picking problem was selected as it allowed for eliciting at least ten requirements will be selected.

Some of the examples of design problems used in other user studies in literature are provided in Appendix B:. Next, the execution procedure for the experiment is discussed in section 5.4

5.4 Execution Procedure

On the day of the experiment, the students were introduced to the study during their regular class time. As previously mentioned the environment in which the experiment is conducted can have significant influence on the experimental results. Familiar environment such as location and time is preferred over unfamiliar environment to avoid any bias and therefore the experiment was conducted in the regular class location and time. The execution of the experiment included the following steps:

1. First, the students were given general instruction about the execution of the experiment.

2. Students were then randomly assigned to three different and equally balanced groups. .
3. Design packets were then distributed to students. These design packets are described in detail in Section 5.5 Once all students have a packet, they were asked to open it and read the instruction sheet. The students were allowed to ask clarification questions. They had five minutes to do so. Pilot study indicates that five minute time period is sufficient for students to read and understand the instructions.
4. A. 0(yes) and 5(yes) groups had 35 minutes to complete the given design task. This included reading the instructions and asking clarification questions (5 minutes), understanding the given problem and eliciting requirements (10 minutes) and generating concepts (20 minutes).

   NE group had 30 minutes to complete the given design task. This included reading instructions (5 minutes), understanding the design problem (5 minutes) and generating concepts (20 minutes). The timeline of the experiment for all three groups is illustrated in Figure 5-2.
5. The students were notified of the remaining time at the end of reading instructions and asking questions, problem understanding and requirements elicitation and idea generation.
6. At the end of idea generation, the students were instructed to stop ideation and complete exit surveys. There was no time limit on the survey; however it was designed for completion in 5 minutes. A sample survey is shown in Appendix C:.

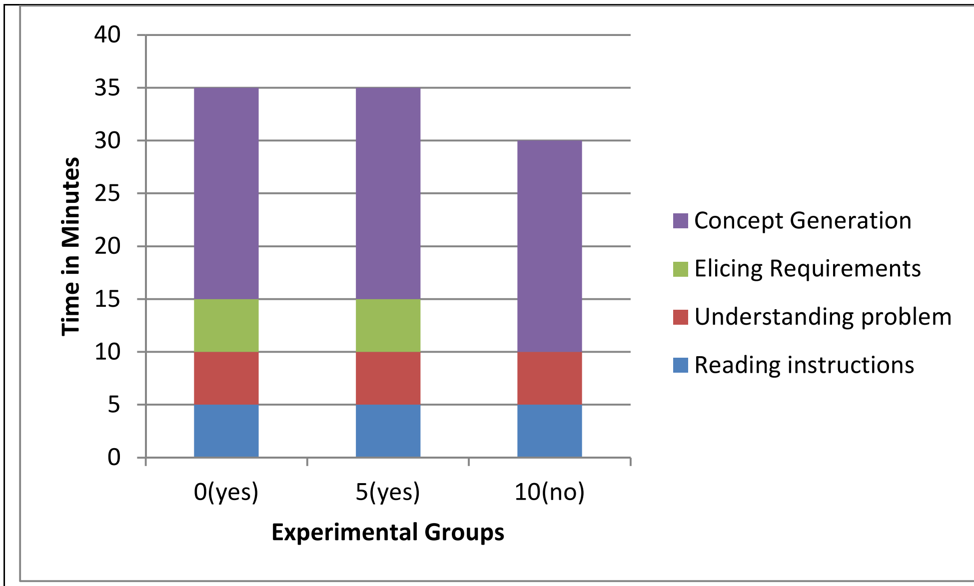


Figure 5-2 Timeline for different experimental groups

The data collected from this experiment will primarily consist of the following:

- List of requirements elicited by students from 0(yes) and 5(yes) groups.
- Solutions generated by the students from all three groups. A sample sheet that provided to the students for documenting their ideas is illustrated in Appendix D:
- Completed exit survey.

A detailed discussion on how this data will be used to answer each sub-question of RQ3 is provided in Section 5.6.

5.5 <u>Details of Design Packet</u>

Each student received a design packet at the beginning of the experiment. This design packet included instruction sheet, details of design task, sheets with space provided for documenting ideas. A survey was provided to each student at the end of the experiment. Each of these items were of a different color so for instance instruction sheet will be blue, problem sheet will be green and so on. This helped to ensure that the students are looking at the correct sheet according to the experiment timeline. Further, students in each group received a different design packet. The specific details of design packet for each group are discussed below.

5.5.1 0(yes) Group

The design packet for 0(yes)-Group included the following:

- A colored instruction sheet- The instruction sheet was uniquely colored to ensure that the students are only looking at the instruction sheet and no other document from the packet. This is critical for the timeline of the experiment. So for instance, in the time allotted for reading the instructions, if the students look at the problem sheet, this will lead to additional time for understanding the problem and ultimately lead to biased results. Therefore to avoid this bias, each of the items in design packet was uniquely colored.
- A colored problem statement and colored requirements checklist to aid requirements elicitation. The goal was to provide some motivation to students for eliciting requirements. Using a checklist will encourage the students to think about different types of requirements and thus a checklist is selected over other tools since it helps to fulfill the goal. A sample requirements elicitation sheet is provided in Appendix E:

- Ten ideation sheets containing concept number, space for concept name, and space for generating ideas. Observations made from pilot study indicate that ten ideation sheets are adequate as none of the students were able to generate more than ten ideas.
- Exit survey

The students were asked to read the instruction sheet first. As per the instructions, the students then read problem statement and elicited requirements for first 10 minutes. The instructions encouraged students to use the checklist but it was not mandated and the students were free to elicit requirements that do not fall under category listed on the checklist. The checklist is provided to help students with the elicitation process.

At the end of 10 minutes, the students started the idea generation on the 'ideation sheet'. The students were free to express their ideas either in words or sketches. Space was provided for students to write their initials. This helped to ensure that the sketches are from the same student. The ideation sheets were sequentially numbered to keep track of number of ideas generated by each student. Each student received ten ideation sheets, but more were provided on request. However, none of the students required more than ten sheets.

At the end of 20 minutes of ideation, the students were asked to end ideation and complete the exit survey. The students were asked to write their initials on the survey. This was done to facilitate the mapping of the survey results to the experiment results.

5.5.2 5(yes) Group

For the partial requirements elicitation group [5(yes) group], the design packet included the following:

- A colored instruction sheet
- A colored problem statement, partial list of requirements and checklist to aid elicitation of more requirements
- 10 ideation sheets containing concept number, space for concept name, and space for generating ideas.
- Exit Survey.

The experimental procedure followed by partial requirements elicitation group was similar to that followed by requirements elicitation group as discussed in Section 5.4. The only difference is that this group will be given some requirements in addition to the problem statement unlike 0(yes) group which had no requirements at all.

5.5.3 10(no) Group

For control group, the design packet included the following:

- A colored instruction sheet
- the problem statement and list of some requirements
- 10 ideation sheets containing concept number, space for concept name, and space for generating ideas.

The students were asked to read the colored instruction sheet first. As per the instructions on colored sheet, the students then read understood the problem statement and requirements. Since the control group students were not required to elicit requirements, they were only given five minutes time to read and understand the problem and requirements. At the end of five minutes, they were instructed to start ideation. As previously mentioned, the students were free to express their ideas either in words or sketches. They had 20 minutes for generating ideas. At the end of idea generation, the students were asked to fill out exit survey.

Next, section 5.6 discusses the analysis of collected data to answer RQ3.

5.6 Data Coding to Support Analysis

After collecting the data from the experiment, the next step was to analyze the data to answer the research question. The collected data consisted of the requirements elicited by the students of 0(yes) and 5(yes) group and the solutions generated by the students of all three groups. In order to analyze this data, it was necessary to refine and code the collected data.

First, the requirements elicited by the students were refined and coded into functional and non-functional. The details of the same are explained in section 5.6.1. Then section 5.6.2 explains the details of extracting the unique requirements against which the solutions were compared. Finally, section 5.6.3 explains the protocol for analyzing the design solutions.

5.6.1 Analysis for Type of Requirements Elicited

The goal of RQ 3.1 is to investigate the type of requirements elicited by novice designers (students) when assigned with the task of eliciting requirements. As a first step for analyzing the typology, the requirements elicited by the students were transcribed from the handwritten text to electronic documents without any editing. However, not all requirements elicited were complete meaningful sentences as several were either missing one or more of a subject, verb, and object. Thus, a protocol was designed to refine the raw data.

5.6.1.1 Protocol for data refinement

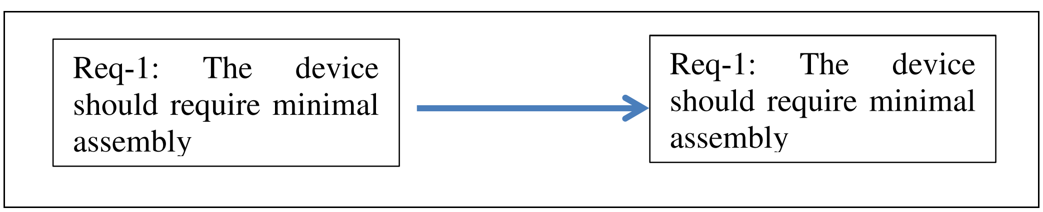
As previously mentioned, the goal of a data refinement protocol is to refine the raw requirements data obtained from the experiment. Here, raw requirements refer to the requirements elicited by the students of 5(yes) and 0(yes) groups. The need for refining the raw requirements arises from the fact that not all student requirements were complete meaningful sentences. Some requirements were phrases, some were questions, and some were compound sentences. Thus, it was necessary to refine and recompose these requirements into complete meaningful sentences so that further analysis could then be performed on that data. Table 5-4 illustrates an example list of student requirement and the corresponding refined list of requirements. Column 2 in Table 5-4 shows student requirements; these are requirements written as-is as elicited by the student. Column 3 in Table 5-4 shows refined requirements; these are requirements derived from student requirements using the data refinement protocol. The protocol for deriving the refined list of requirements is discussed next using each requirement in Table 5-4 as example.

Table 5-4 Example of requirements

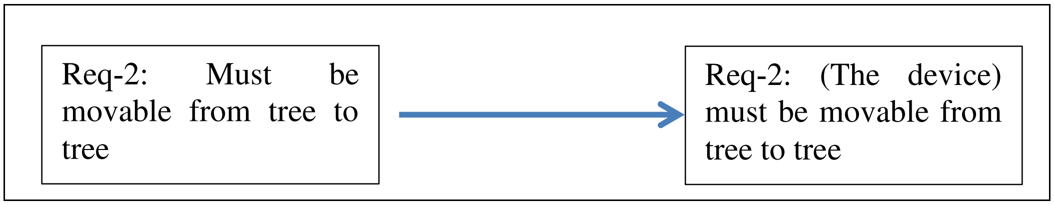
	Student requirement	Refined list
Req-1	The device should require minimal assembly	The device should require minimal assembly
Req-2	Must be movable from tree to tree	(The device) must be movable from tree to tree
Req-3	Force	Word –not considered
Req-4	The device must require 1PM per year	Ambiguous –Not considered
Req-5	Must operate through harvest season with minimal maintenance	(The device) must operate through harvest season
		(The device must require) minimal maintenance
Req-6	Is the device light weight?	The device must be light weight

The following steps were followed for refining the data:

- The requirements that were complete sentences were considered as-is for the analysis. For example, the requirement "The device should require minimal assembly" is a complete sentence, thus it was considered as is in the refined list.



- Some requirements were elicited as phrases. These phrases were re-written to form complete sentences by adding appropriate subjects or objects. So for instance, the requirement "Must be movable from tree to tree" was re-written as - (The device) must be movable from tree to tree. Here, the subject 'The device' was added to complete a meaningful sentence.



- Some students only had single words as requirements such as “forces” or “materials”. While the words give a general idea that the students were thinking about those categories, they do not specifically state what the requirement is. Further, the requirements that are only words cannot be classified into functional or non-functional. Additionally, there were other requirements that were ambiguous. Thus, these “requirements” were not considered for this analysis.
- Since one of the goals of this study is to compare the number of functional and non-functional requirements elicited between 5(yes) and 0(yes) groups, it was essential to split the compound requirements. This will ensure that the numbers of requirements are counted correctly. Thus, the compound requirements were split into simple requirements. So for instance the requirement –“Must operate through harvest season with minimal maintenance” was re-written as two different requirements – 1) “(The device) must operate through harvest season” and 2) “(The device) must operate with minimal maintenance”.

Req-5: Must operate through harvest season with minimal maintenance.	→	Req-5a: (The device) must operate through harvest season Req-5b: (The device) must operate with minimal maintenance

It may be noted that while splitting the compound requirements, some contextual information may be lost, but this loss is out of scope for this research. So, in the above example, when the requirement elicited by the student is split in two different requirements, the relation between the two requirements is lost as the requirement – '(The device) must operate through harvest season' does not indicate that the device must also require minimal maintenance while doing so. This was the original intent of the requirement elicited by the student which is lost when the requirement is split.

- Some students had requirements framed as questions. These questions were restructured as requirement sentences.

For example, requirement such as – "Is the device lightweight?" was restructured as requirement statement and re-written as "The device (must be) light weight".

Req-6: Is the device lightweight?	→	Req-6: The device must be light weight.

Following the data refinement protocol, all requirements elicited by students of 5(yes) and 0(yes) groups were re-written to form the refined list of requirement. Table

5-5 illustrates the summary of number of each transformation for 5(yes) and 0(yes) group.

Table 5-5 Summary of requirements transformation

Group	Completions	Deletions	Compound split	Questions Reformulated
5(yes)	112	2	30	0
0(yes)	117	11	35	11

Each requirement in the refined list was then coded into functional and non-functional. The protocol for coding the requirements into functional and non-functional is discussed in Section 5.6.1.2.

5.6.1.2 Protocol for coding into functional and non-functional

The goal of RQ 3.3 is to compare the number of functional and non-functional requirements elicited by students of 5(yes) and 0(yes) group. Thus, a protocol was developed to code the refined requirements in to functional and non-functional. Requirements were classified as functional requirements if they satisfied one of the following definitions or rules:

- The requirement indicates what the system or product must do. The requirement includes transformative or active verb. For example-***the device must pick at least two fruits at a time***. Here the requirement indicates what the device must do – "pick at least two fruits" and thus it is coded as functional.
- The requirement indicates the functions that system performs on itself. For example, ***the device must collapse to fit in storage space*** indicates that the device

"must collapse" which is a function that the device performs on itself. Thus, the requirement is be coded as functional.

- The requirement indicates the functions that system must not do. For example, ***the device must not damage the fruit*** indicates a function that the device must not do –"must not damage fruit" and is thus coded as functional.
- Requirement indicates functions that user can perform using the system. For example, ***the device must allow user to pick multiple fruits at a time*** indicates a function that user can perform, "must allow user to pick", and is thus coded as functional.
- The requirement indicates functions that the user can or cannot perform. For example, ***the user must not leave the wheel-chair*** indicates that the user cannot "leave wheel-chair", a function, and thus it is coded as functional.

While a requirement that is not coded as functional could be assumed to be non-functional, a separate coding scheme is provided to ensure that requirements are independently non-functional requirements. The requirements were coded as non-functional if they satisfied at least one of the following definitions or rules:

- The requirement defines the existence of the system or product. For example, ***the design must have two wheels*** defines what the system must have and this is coded as non-functional.

- The requirement defines the properties of the system or product. For example, ***the device must fit in 4 feet X 3 feet storage*** defines a size property for the system.
- These requirements typically indicate something that the device must/must not have. For example, ***the device must not have sharp edges*** describes properties that are excluded from the solution. Alternatively, ***the device must not be harmful to the user*** describes a characteristic of the system.
- The requirement indicates the characteristics the user must have. For example, ***the user must have upper body strength*** indicates the characteristic of "upper body strength" that the user must have in order for proper operation and thus would be coded as non-functional.

Requirements elicited by students of 0(yes) and 5(yes) groups were classified into functional and non-functional following the protocol discussed in Section 5.6.1.2. All requirements were classified by the researcher. In order to verify the robustness of the protocol, two additional raters classified 54 requirements into functional and non-functional. Inter-rater reliability was tested using Joint probability agreement and agreement between the raters is AB= 0.81, AC = 0.85, BC = 0.72 and ABC = 0.7. An agreement of 0.7 is considered as acceptable and since all the values are above 0.7, the protocol of classifying into functional and non-functional is considered robust and further analysis is performed.

5.6.2 Protocol for Extracting Unique Requirements

In order to compare the solutions against requirements elicited by students, a list of unique requirements was created. In this study, the students of 0(yes) and 5(yes) groups elicited requirements. However, there was some overlap in the requirements elicited by the students. A protocol was developed to eliminate the repeated requirements and while creating a list of unique requirements. The solutions can then be compared against this list of unique requirements to understand the effect of designer involvement in problem definition on the extent of requirements addressed.

A total of 167 requirements were elicited by students of 5 (yes) group while the students of 0(yes) group elicited a total of 182 requirements. The requirements elicited by students were listed in evaluation sheet. Several strategies were used to identify similar requirements and eliminate them so that a list of unique requirements could be created. These strategies are discussed below:

- Same requirements – Requirements elicited by some students were exactly the same. These requirements were grouped together and only one of the requirements from each group was considered.
- Key word identification – Key words were used to identify similar requirements. The requirements having the same key words were grouped together and only one requirement from the requirements with same key word was considered in the list of unique requirements. For example, consider the requirements ***device must allow for use of upper body strength*** and ***forces to operate the device must be within upper body strength of a person***. These two requirements are considered

similar with “upper body strength” as key word phrase and only one of the two requirements is considered in the list of unique requirements. It may be noted that the key words were considered within context. Thus, the requirements having the same key words were also compared to see that they were conveying the same meaning.

- Synonyms – Another strategy used to identify similar requirements is comparing requirements for synonyms. For example, consider the requirements ***the device must be manually operated*** and ***the device must be human powered***. These two requirements are similar with ‘manually operated’ and ‘human powered’ as synonymous. Again, the synonyms were considered within context and the requirements were compared to make sure that they convey the same meaning.
- While considering the unique requirements, target values were not used to differentiate requirements. Thus, the requirement ***the device must cost 50 dollars*** was considered similar to the requirement ***the device must cost 20 dollars*** and only one of the two requirements was in the list of unique requirement. Here ‘cost’ is the critical aspect. It may be noted that the solutions are analyzed for whether or not a particular requirement is addressed, not how well it the solution could meet the target value. Further, whether or not a requirement is met is not considered for this study and reserved for future research. Thus, the target values are not critical to this research.

Considering the strategies discussed above, the requirements with same key words or synonyms were grouped together. Table 5-6 shows the number of reductions made using each of these strategies for 5(yes) and 0(yes) group.

Table 5-6 Summary of number of strategies applied

Group	Keyword	Synonyms	Exact Same
5(yes)	66	13	8
0(yes)	57	23	12

Following the above strategies, the number of unique requirements extracted from 5(yes) and 0(yes) group are respectively seventy five and eighty. After extracting unique requirements for 0(yes) and 5(yes) groups, the two lists was compared following above strategies to extract the unique requirements. Thus, a list of 123 unique requirements was extracted from the requirements elicited by students of 0(yes) and 5(yes) groups. A complete list of unique requirements can be found in Appendix F:

5.6.3 Protocol for analyzing design solutions

The aim of RQ 3.2 is to investigate whether or not designer's involvement in eliciting requirements affects the number of requirements addressed in the design solutions. In order to be able to do so, it was necessary to identify the means in the solutions developed by the students that address a particular requirement. Again, these are means that address the requirement but that may or may not meet the requirement. As shown in previous research, low levels of fidelity of representations are suitable for determining whether a requirement might be met by the solution, but they cannot support the converse in that the requirement is not met [81, 82]. Therefore, the addressment condition is used, rather than trying to evaluate how well a concept meets a requirement.

For this designer study, the students are not limited to sketches in documenting their solutions, thus both sketches and textual descriptions are considered for analysis. The solution must explicitly address at least one requirement to be considered for further analysis. So the solutions such as "Pay a person to pick fruit" or "beaver" were not considered for analysis. In the study, the students were explicitly asked to limit one idea per sheet, thus multiple sketches on the same sheet were considered as one solution. Further, if the student had the same idea with more details drawn on new sheet, it was considered as a different solution.

A total of 155 solutions were analyzed against 128 unique requirements. For analyzing the solution, an evaluation table is used (Figure 5-3) for tracking and coding purposes.

The first column of the table indicates the requirement code number. The requirement code indicates whether a requirement was embedded in the problem, given in the list or elicited by the student. Thus, P2, P3 etc. refer to the requirements embedded in the problem statement. The second column captures the requirement itself. The solution code indicates the identification of the solution to be analyzed against each requirement. This identification number corresponds to unique student identifier and the sketch number. So, the identification code of 5(Y)-B-3 refers to sketch number 3 for the student B of 5(yes) group.

Requirement Code	Requirements	Solution Code		
		5(Y)-A-1	5(Y)-B -3	…
		Means	Means	
P2	The device must allow the disabled in wheel chair to collect peaches in basket while still in the wheel chair			
P3	The device must be manually operated.			
P4	The device must prevent damage to the fruit while picking it.			
P5	The fruit should not fall on the ground while picking			
……..				

Figure 5-3: Snap shot of tracking table for solution analysis.

The solution was compared against the requirement and the means was populated in the cells of sheet. Here means refers to the solution fragment that addresses a requirement. Again, since the students were not limited to sketching, the solution fragment could be a sketch fragment or text fragment. If a solution did not have a means to address a particular requirement, '0' was entered in the cell. Figure 5-4 shows snap shot of completed tracking table with means populated for a solution.

Requirement Code	Requirements	Solution Code
		5(Y)-B -3
		Means
P2	The device must allow the disabled in wheel chair to collect peaches in basket while still in the wheel chair	Sliding Pole
P3	The device must be manually operated.	String mechanism
P4	The device must prevent damage to the fruit while picking it.	Mesh cup
P5	The fruit should not fall on the ground while picking	Mesh cup
……..		

Figure 5-4 Snap shot of completed tracking table.

In addition to populating the means in the tracking tables, the means were also circled on the sketch and the requirement code that the means addressed was written. Figure 5-5 shows an example of analyzed solution sketch. It may be noted that the means populated the tracking table in Figure 5-4 are circled in the sketch.

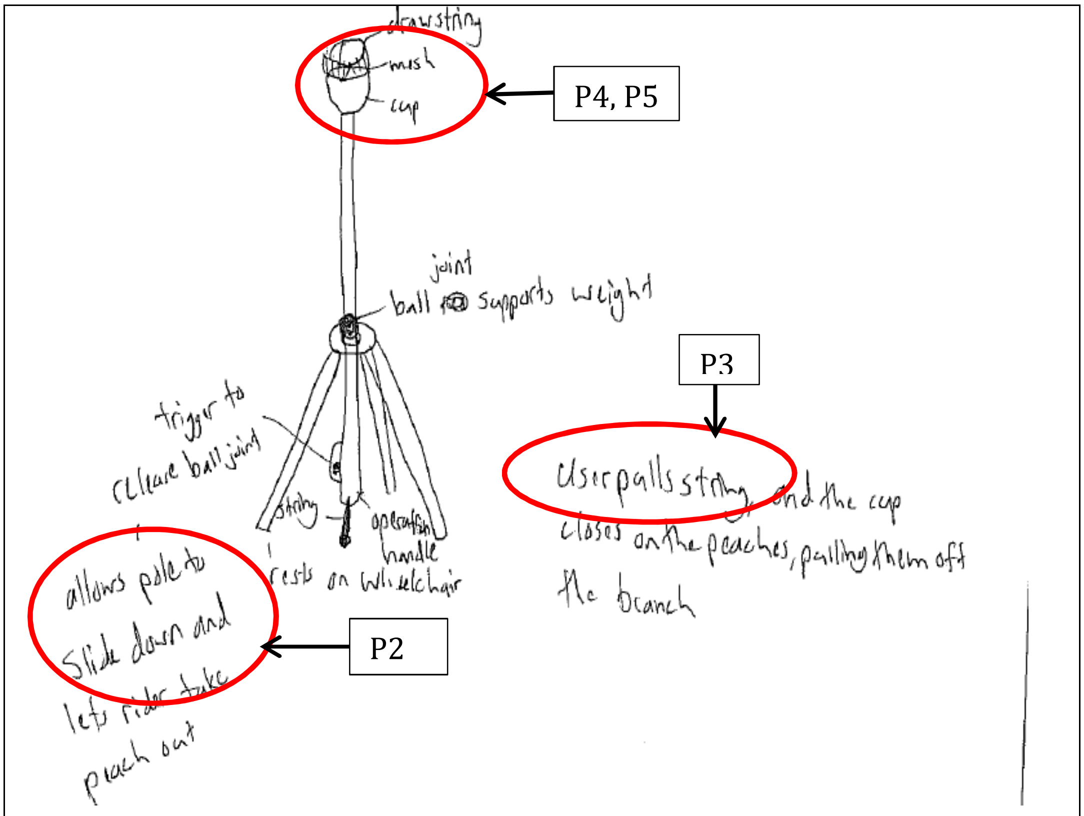


Figure 5-5 Example of analyzed solution

To ensure that all the sketches are analyzed uniformly, an analysis protocol was created to identify means for each of 128 requirements. Figure 5-6 illustrates a snap shot of guidelines for identifying means. This detailed protocol can be found in Appendix F:. The protocol was followed while populating the means for all 155 sketches. After identifying means addressing the requirements for all sketches, number of requirements addressed per sketch was counted. These numbers are then used to compare for the extent of requirements addressed for each group.

Requirement-code	Requirements (Embedded in problem statement or given to students)	Guidelines for identifying means
P1	The device must allow the disabled in wheel chair to experience the joy of picking peaches from the tree.	The solution must have indication of allowing user to pick the fruit from tree. For example a ramp that allows user to touch fruit
P2	The device must allow the disabled in wheel chair to collect peaches in basket while still in the wheel chair	The solution must indicate that the user does not require to get out of the chair in order to pick the fruit. For example extendable arms that allow to pick fruit,
P3	The device must be manually operated.	The solution must have indication of use by hand such as lever or gear system, must not have automation
P4	The device must prevent damage to the fruit while picking it.	The solution must not have sharp edges or grips that could damage the fruit.
P5	The fruit should not fall on the ground while picking	The solution must indicate a means that prevents the fruit from falling to the ground. This could some sort or guard or basket that prevents the fruit from falling

Figure 5-6 Snap shot of guidelines for identifying means

In order to test the robustness of the protocol, inter-rater reliability test was done using joint probability agreement. Two additional raters coded three sketches against all unique requirements and agreement between the raters was AB = 0.66, AC = 0.70 and BC=0.66. For the purpose of this research, value of 0.6 was considered acceptable. It may be noted that this value is on the lower side of the acceptable range, thus there is a scope of improvement in the protocol. The findings from this analysis are discussed in Section 5.8.

5.7 Findings from type of requirements (RQ3.1)

RQ 3.1 aims at investigating the type of requirements elicited by novice designers when assigned with task of eliciting requirements. Two types of requirements are considered here –functional and non-functional. After coding the requirements elicited by the students of 0(yes) and 5(yes) groups into functional and non-functional, the number of functional and non-functional requirement per student was counted. An ANOVA was conducted to see if there is a difference in the average number of requirements between at least one of the pairs while comparing 0(yes) functional, 0(yes) non-functional, 5(yes) functional, and 5(yes) non-functional). The level of significance of 0.05 was used for this study and p-value obtained from ANOVA was compared to the level of significance. It may be noted that there are three assumptions for conducing ANOVA study. These are 1) independence, 2) normality and 3) homogeneity of variance. However, for this research, the study was designed such that the participants were randomly selected for each condition. Thus, these assumptions are not explicitly tested for the study. The ANOVA results from this comparison are illustrated in Table 5-7. A p-value of 0.001445 suggests that there is significant difference in the average number of requirements per student between at least one of the four groups. Further, comparing the averages, it can be observed that the average number of non-functional requirements elicited per student is higher than the average number of functional requirements elicited per student in both 0(yes) and 5(yes) groups.

Table 5-7 ANOVA comparing 0(yes) functional, 0(yes) non-functional, 5(yes) functional and 5(yes) non-functional

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
5(yes)-F	16	50	3.125	3.983333
5(yes)-NF	16	115	7.1875	12.9625
0(yes)-F	15	56	3.733333	7.495238
0(yes)-NF	15	114	7.6	32.82857

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	248.1985	3	82.73284	5.860978	0.001445	2.763552
Within Groups	818.7208	58	14.11588			
Total	1066.919	61				

In order to identify which of the average values are significantly different, further analysis was conducted. This is discussed in sections 5.7.1 and 5.7.2

5.7.1 Comparing for functional versus non-functional requirements elicited

An ANOVA was conducted to determine if there is a significant difference in the average number of functional and non-functional requirement elicited per student within 5(yes) group. Table 5-8 shows ANOVA results from this comparison.

A p-value of 0.000441 is observed, which is lower than the level of significance. This leads to an inference that there is a significant difference in the average number of functional and non-functional requirements elicited per student within the 5(yes) group. Further, looking at the average values, the average number of non-functional requirements elicited per student is greater than the average number of functional requirements elicited per student.

Table 5-8 Comparing functional versus non-functional in 5(yes) group

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
5(yes)-F	16	50	3.125	3.983333		
5(yes)-NF	16	115	7.1875	12.9625		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	132.0313	1	132.0313	15.58274	0.000441	4.170877
Within Groups	254.1875	30	8.472917			
Total	386.2188	31				

Table 5-9 illustrates the ANOVA results for comparing functional and non-functional requirements elicited within 0 (yes) groups.

Table 5-9 Comparing functional versus non-functional in 0(yes) group

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
0(yes)-F	15	56	3.733333	7.495238		
0(yes)-NF	15	114	7.6	32.82857		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	112.1333	1	112.1333	5.561644	0.025574	4.195972
Within Groups	564.5333	28	20.1619			
Total	676.6667	29				

A p-value of 0.025574 is observed, which, again, is lower than the level of significance of 0.05. This leads to conclusion that there is a significant difference in the average number of functional and non-functional requirements elicited per student within 0(yes) group. Further, comparing the average values, the average number of non-functional requirements elicited per student is greater than the average number of functional requirements elicited per student.

Thus in both 0(yes) and 5(yes) groups, the average number of non-functional requirements elicited per student is greater than the average number of functional requirements elicited per student. It may be noted that while the students of 5(yes) groups were given list of five requirements in addition to the design problem, the students of 0(yes) group were only given the design problem. However, in both cases, it is found that the average number of non-functional requirements elicited is higher than the average number of functional requirements. Thus, providing a list of requirements in addition to asking the students to elicit more requirements did not have additional benefits while eliciting requirements. Fewer numbers of functional requirements could result from the fact that functions of a system are more obvious and thus the students might miss to elicit them as important requirements. This, then means that more emphasis needs to be placed on functional requirements specially in the elicitation phase as students are more likely to miss them.

Next, section 5.7.2 shows the comparison between 5(yes) and 0(yes) groups for the average number of functional and non-functional requirements.

5.7.2 Comparing 5(yes) and 0(yes) groups

Table 5-10 illustrates ANOVA results for comparing the average number of functional requirements generated between 5(yes) and 0(yes) groups.

Table 5-10 Comparing 5(yes) and 0(yes) group for functional requirements

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
5(yes)-F	16	50	3.125	3.983333		
0(yes)-F	15	56	3.733333	7.495238		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	2.865054	1	2.865054	0.504523	0.483193	4.182964
Within Groups	164.6833	29	5.678736			
Total	167.5484	30				

A p-value of 0.483193 indicates that there is no significant difference in the average number of functional requirements elicited per student between the 0(yes) and 5(yes) group. It may be noted that the students of 5(yes) group were provided with a list of five requirements while the students of 0(yes) group were only given the problem statement. However, the performance of the students was identical in terms of eliciting functional requirements. Thus, providing a list of requirement to the students of 5(yes) group did not influence the number of functional requirements elicited by the two groups.

Table 5-11 illustrates ANOVA results for comparing the average number of non-functional requirements elicited between 5(yes) and 0(yes) groups.

A p-value of 0.810 indicates that there is no significant difference in the average number of non-functional requirements elicited per student between the 5(yes) and 0(yes) groups. It may be noted that the students of 5(yes) group were provided with a list of five requirements while the students of 0(yes) group were only given the problem statement. However, the performance of the students was identical in terms of eliciting functional requirements. Thus, providing a list of requirements to the students of 5(yes) group did not influence the number of functional requirements elicited by the two groups.

Table 5-11 Comparing 5(yes) and 0(yes) group for non-functional requirements

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
5(yes)-NF	16	115	7.1875	12.9625		
0(yes)-NF	15	114	7.6	32.82857		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	1.317339	1	1.317339	0.058411	0.810727	4.182964
Within Groups	654.0375	29	22.55302			
Total	655.3548	30				

After comparing the functional and non-functional requirements elicited by the students, the next step was to compare the functional and non-functional requirements addressed in the solutions. Section 5.7.3 discusses the findings from comparing the number of functional and non-functional requirements addressed by 5(yes) and 0(yes) groups.

5.7.3 Comparing functional and non-functional requirements addressed

After comparing the functional and non-functional requirements elicited by the students of 5(yes) and 0(yes) group, the next step was to compare the functional and non-functional requirements addressed by each group. In order to do this comparison, the requirements addressed by each student from 5(yes) and 0(yes) group were coded into functional or non-functional by following the protocol established in section 5.6.1.2. It is hypothesized that while the students of both the groups have elicited more non-functional requirements compared to functional requirements, the ratio non-functional/functional will decrease while considering the requirements addressed. It is easier to comprehend the functions of the system (functional requirements) compared to characteristics of the system (non-functional requirements) and thus, students will have more functional requirements addressed in the final solutions.

Section 5.7.3.1 discusses the findings from comparing functional and non-functional addressed in 5(yes) group while section 5.7.3.2 discusses the findings from comparing functional and non-functional requirements in 0(yes) group.

5.7.3.1 Comparing functional and non-functional addressed in 5(yes) group

Table 5-12 illustrates the findings from comparing the functional and non-functional requirements addressed by the students of 5(yes) group while considering each solution as a data point.

Table 5-12 Comparing functional and non-functional addressed in 5(yes) – Individual

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
5(yes) F-addressed	59	97	1.644068	2.0263		
5(yes) NF-addressed	59	136	2.305085	4.42256		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	12.88983	1	12.88983	3.997553	0.047904	3.922879
Within Groups	374.0339	116	3.22443			
Total	386.9237	117				

A p-value of 0.047904 indicates that there is a difference in the average functional and non-functional requirements. Further, on average, more non-functional requirements are addressed compared to functional requirements.

Table 5-13 illustrates the findings from comparing the functional and non-functional requirements addressed by 5(yes) group while considering each student as data point.

Table 5-13 Comparing functional and non-functional addressed in 5(yes)-Integrated

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
5(yes)-INT-F-addressed	16	42	2.625	3.05
5(yes)-INT NF-addressed	16	63	3.9375	6.0625

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	13.78125	1	13.78125	3.024691	0.092259	4.170877
Within Groups	136.6875	30	4.55625			
Total	150.4688	31				

A p-value of 0.092259 indicates that there is no difference in the average number of functional and non-functional requirements addressed. It can also be observed that the average number of non-functional requirements addressed is only slightly more when compared to the average non-functional requirements addressed.

Table 5-14 illustrates the summary of functional and non-functional requirements elicited and addressed by 5(yes) group.

Table 5-14 Summary of functional and non-functional elicited and addressed

5(yes)	Elicited	Addressed (Individual)	Addressed (Integrated)
Functional	3.125	1.644	2.625
Non-Functional	7.1875	2.3050	3.9375
p-value	0.000441	0.0479	0.09225

It can be observed that on an average more non-functional requirements are elicited compared to functional requirements. However the ratio of average number of non-functional to functional decreases when considering the requirements addressed.

Further, the difference between the average functional and non-functional requirements elicited is statistically significant with a p-value of 0.000441, while the difference is not significant when considering the average functional and non-functional requirements addressed by considering each student as a data point (p-value of 0.09225 > alpha- 0.05). This illustrates that although the students elicit more non-functional requirements than functional requirements, the performance while addressing the requirements is identical. One possible explanation for this is that the functions of the systems are more obvious and thus the students might miss to elicit them as requirements. However, this is reflected while analyzing the addressed requirements as the ratio of the number of non-functional requirements to the number of functional requirements decreases. This shows that the students are able to comprehend the functional requirements better than the non-functional requirements.

5.7.3.2 Comparing functional and non-functional addressed in 0(yes) group

Table 5-15 illustrates the findings from comparing the average number of functional and non-functional requirements addressed by 0(yes) group while considering each solution as a data point.

Table 5-15 Comparing functional and non-functional addressed in 0(yes) – Individual

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
0(yes)-F-addressed	60	136	2.266667	4.164972
0(yes)- NF-addressed	60	170	2.833333	7.361582

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	9.633333	1	9.633333	1.671503	0.198583	3.921478
Within Groups	680.0667	118	5.763277			
Total	689.7	119				

It can be observed that there is no significant difference in the average number of functional and non-functional requirements addressed by 0(yes) group while considering each solution as a data point as indicated by a p-value of 0.198583.

Table 5-16 illustrates ANOVA results for comparing the average number of functional and non-functional requirements addressed in 0(yes) group while considering each student as a data point.

Table 5-16 Comparing functional and non-functional addressed in 0(yes) – Integrated

SUMMARY

Groups	Count	Sum	Average	Variance
(RE-INT)-F-addressed	15	43	2.866667	5.838095
(RE-INT) NF-addressed	15	63	4.2	16.45714

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	13.33333	1	13.33333	1.19607	0.283428	4.195972
Within Groups	312.1333	28	11.14762			
Total	325.4667	29				

Again, a p-value of 0.283428 indicates that there is no significant difference in the average number of functional and non-functional requirements addressed by students of 0(yes) group while considering each student as a data point.

Further, Table 5-17 illustrates a summary of average number of functional and non-functional requirements elicited and addressed by 0(yes) group.

Table 5-17 Summary of functional and non-functional addressed and elicited

0(yes)	Elicited	Addressed (Individual)	Addressed (Integrated)
Functional	3.7333	2.2666	2.8666
Non-Functional	7.6	2.8333	4.2
p-value	0.0255	0.1985	0.2834

It can be observed that on an average more non-functional requirements are elicited compared to functional requirements. However the ratio of average number of non-functional to functional decreases when considering the requirements addressed.

The difference between the average number of functional and non-functional elicited by 0(yes) group is significant with a p-value of 0.0255. While the difference between the average number of functional and non-functional requirement is not significant in both cases, while considering each solution as a data point (p-value - 0.1985) and while considering each student as a data point (p-value -0.2834).

This is similar to the observation made for 5 (yes) groups. Again, this validates that the students are able to comprehend functional requirements better than non-functional requirements.

Thus, by comparing the average number of functional and non-functional requirements elicited and addressed, for both 0(yes) and 5(yes) groups it is found that though more non-functional requirements are elicited compared to functional requirements, the ratio of number of non-functional requirements to functional requirements decreases when considering the addressed requirements. In other words, more number of non-functional requirements are not addressed compared to the number of functional requirements.

This leads to the recommendations that the students should be explicitly asked to focus on the functional requirements while eliciting the requirements as there is a greater likelihood of missing to elicit them. However, they should be asked to focus on non-functional requirements while addressing requirements as there is a greater likelihood of missing to address them.

Next, section 5.8 discusses the findings for whether or not designer’s involvement in requirements elicitation influences the number of requirements addressed in the solution.

5.8 Findings from Solution Analysis (RQ3.2)

A total of 155 solutions were analyzed against 128 unique requirements to investigate designer involvement in generating requirements and extent of requirements addressed. ANOVA was conducted to compare the extent of requirements addressed between the three groups. It may be noted here that since the students were asked to document their solution ideas in either sketch or textual form, the solutions included either sketches or textual descriptions of the ideas. Further, regardless of the groups, all the solutions were compared against a list of 128 unique requirements derived by the union of requirements embedded in the problem statement, requirements from the list given to the students and the requirements elicited by students of partial elicitation [5(yes)] and elicitation [0(yes)] groups. An evaluation table was created with first column as requirements and first row as sketches. The cells were filled with means for each requirement. The protocol for analyzing the solutions is described in Section 5.6.3. This evaluation table was then used to extract the number of given and unique requirements addressed by each sketch. The numerical values were then used to make comparison between groups to investigate the number of given and unique requirements addressed.

Further, two types of comparison were done between the three groups– 1) Individual - considering each sketch as a data point and 2) Integrated - considering each student as a data point. In order to conduct integrated comparison where each student was considered as a data point, all requirements met by all sketches of a student were integrated into one. For instance, if student A had three sketches and sketch 1 met requirements 1, 2, 3, sketch 2 met requirements 2, 4, 5, 6 and sketch 3 met requirements 3, 6,7, 8, the integrated requirements met by student A would be requirements 1, 2, 3, 4, 5, 6, 7 and 8. A comparison of integrated values was done to mitigate the effect of different number of sketches generated by each student. Table 5-18 summarizes the different comparisons made between the three groups.

Table 5-18 Summary of comparison for given and unique requirements

Requirements	Type	Comparison	Difference between Averages	Comparing Averages
Given (Problem +List of given requirements)	Individual Section 5.8.1	5(yes), 0(yes), 10(no)	Significant	10(no)>0(yes)>5(yes)
		5(yes) vs 0(yes)	Significant	0(yes)>5(yes)
		0(yes) vs 10(no)	Not Significant	
		10(no) vs 5(yes)	Significant	10(no)>5(yes)
	Integrated Section 5.8.2	5(yes), 0(yes), 10(no)	Not significant	
Unique (Given +Student elicited)	Individual Section 5.8.3	5(yes), 0(yes), 10(no)	Significant	10(no)>0(yes)>5(yes)
		5(yes) vs 0(yes)	Not significant	
		0(yes) vs 10(no)	Significant	10(no)>0(yes)
		10(no) vs 5(yes)	Significant	10(no)>5(yes)
	Integrated Section 5.8.4	5(yes), 0(yes), 10(no)	Not significant	

5.8.1 Findings from comparing for given requirements (individual)

Analysis of Variance (ANOVA) was performed to investigate if there is a significant difference in the average number of given requirements addressed between the three groups [10(no), 5(yes) and 0(yes)]. Here, the given requirements include the requirements embedded in the problem statement (5 requirements) and requirements from the list given to the students (10 requirements). Table 5-19 shows the ANOVA results for comparing the three groups for addressing given requirements while considering each solution as a data point.

Table 5-19 ANOVA table for given requirements (Individual)

Groups	*Count*	*Sum*	*Average*	*Variance*		
[5(yes)]	59	377	6.389831	5.138515		
[0(yes)]	60	452	7.533333	4.524294		
[10(no)]	36	294	8.166667	3.628571		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	78.71664	2	39.35832	8.645589	0.000278	3.055558
Within Groups	691.9672	152	4.552416			
Total	770.6839	154				

A p-value of 0.000278 was found indicating that there is a significant difference in the average number of given requirements addressed between at least two of the three groups. Further, looking at the average values between the three groups, 5 (yes) group has the least average given requirements addressed among the three groups, while 10(no) has the highest average given requirements addressed.

In order to further investigate which of the two pairs from the group differ significantly, ANOVA was conducted between three possible pairs from the above groups. These are – 1) 5(yes) vs. 0(yes), 2) 0(yes) vs. 10(no) and 3) 10(no) vs. 5(yes). Table 5-20 shows ANOVA results for comparing between 5(yes) and 0(yes) groups.

Table 5-20 ANOVA for 5(yes) vs. 0(yes) for given requirements (Individual)

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
5(yes) -(Given)	59	377	6.389831	5.138515		
0(yes)-(Given)	60	452	7.533333	4.524294		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	38.89831	1	38.89831	8.055516	0.00535	3.922173
Within Groups	564.9672	117	4.82878			
Total	603.8655	118				

A p-value of 0.00535 indicates that there is a significant difference in the average number of given requirements addressed between 5(yes) and 0(yes) groups. Further, looking at the average number of requirements addressed between the two groups, 0(yes) has a greater average given requirements addressed compared to 5(yes) group. While the students of both 0(yes) and 5(yes) groups were tasked with eliciting requirements before developing solution ideas, the students of 5(yes) group were given a list of 5 requirements. The ANOVA results suggest that giving the students a list of requirement seems to negatively influence the number of requirements addressed in the solutions.

Table 5-21 illustrates the ANOVA results for comparing between 0(yes) and 10(no) groups for average number of given requirements addressed.

A p-value of 0.14558 indicates that there is no significant difference between the average numbers of given requirements addressed between the two groups. In terms of the experiment condition, 0(yes) and 10(no) groups were quite different from each other. 0(yes) group was not given any requirement (other than problem statement), while 10(yes) group was given a list of 10 requirements.

Table 5-21 ANOVA for 0(yes) vs. 10(no) for given requirements (Individual)

Groups	*Count*	*Sum*	*Average*	*Variance*		
0(yes)-(Given)	60	452	7.533333	4.524294		
10(no)-(Given)	36	294	8.166667	3.628571		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	9.025	1	9.025	2.153537	0.14558	3.942303
Within Groups	393.9333	94	4.19078			
Total	402.9583	95				

While 0(yes) group elicited their own requirements, 10(no) group did not elicit any requirement. However, there is not much difference in the number of given requirements addressed between the two groups.

Table 5-22 illustrates the ANOVA results for comparing average number of given requirements addressed between 10(no) and 5(yes) groups.

Table 5-22 ANOVA for 10(no) vs. 5(yes) for given requirements (Individual)

Groups	*Count*	*Sum*	*Average*	*Variance*		
10(no)-(Given)	36	294	8.166667	3.628571		
5(yes)-(Given)	59	377	6.389831	5.138515		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	70.58715	1	70.58715	15.4449	0.000163	3.943409
Within Groups	425.0339	93	4.570257			
Total	495.6211	94				

A p-value of 0.000163 was found leading to the inference that there is a significant difference in the average requirements addressed between 10(no) and 5(yes) groups. Further, looking at the average values, 10(no) had greater number of average requirements addressed compared to 5 (yes) groups.

Looking at the results in Table 5-20, Table 5-21 and Table 5-22, 5(yes) group has significantly low average given requirements addressed compared to 0(yes) and 10(no) groups. The students of 5(yes) group were given list of five requirements and were also asked to elicit more requirements before developing solution ideas. It is interesting to note that the students who were not given any requirements [0(yes) group] and the students who were given requirements and did not elicit any requirements [10(no) group] performed similar in terms of average given requirements addressed in the solutions. While the students who were given requirements and also had to elicit their own requirements [5(yes)] performed poorly in terms of average given requirements addressed in the solutions. This shows that there are no potential benefits of giving the students a list of requirements while also asking them to elicit their own requirements.

Next, section 5.8.2 discusses the findings from comparing the average number of given requirements addressed while considering each student as a data point.

5.8.2 Findings from comparing number of given requirements addressed (Integrated)

ANOVA was conducted to investigate if there is significant difference in the average number of given requirements addressed between three groups while considering each student as a data point. By considering each student as a data point, the effect of different number of solutions developed by each student can be mitigated. Table 5-23 illustrates ANOVA table for comparing given requirements addressed while considering each student as a data point.

Table 5-23 ANOVA for given requirements (Integrated)

Groups	Count	Sum	Average	Variance		
5(yes)-INT-GIV	16	163	10.1875	2.429167		
0(yes)-INT-GIV	15	168	11.2	4.314286		
10(no)-INT-GIV	14	148	10.57143	2.263736		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	8.04504	2	4.02252	1.338014	0.273319	3.219942
Within Groups	126.2661	42	3.006335			
Total	134.3111	44				

A p-value of 0.273319 indicates that there is no significant difference in the average given requirements addressed between three groups when considering each student as a data point. This is contrary to the observations made while comparing the number of given requirements by considering each solution as a data point.

While there was significant difference in the average number of given requirements addressed between the three groups, there is no difference while considering each student as a data point. This shows that on average, each student is able to address same number of given requirement irrespective of whether they are eliciting the requirements or not. Thus, while considering each student as a data point, the act of eliciting their own requirements does not seem to have any effect on the average given requirements addressed.

After comparing for the given requirements addressed, the next step was to compare the unique requirements addressed in the solutions. The findings from this comparison are discussed in section 5.8.3.

5.8.3 Findings from comparing unique requirements (Individual)

Analysis of Variance (ANOVA) was performed to investigate if there is a significant difference in the average number of unique requirements addressed between the three groups [10(no), 5(yes) and 0(yes)]. A total of 128 unique requirements, obtained by union of requirements embedded in problem statement, list of given requirements and requirements elicited by students of 5(yes) and 0(yes) groups were considered here. Table 5-24 illustrates ANOVA table for comparing for unique requirements considering each solution as a data point (individual).

Table 5-24 ANOVA table for comparing for unique requirements (Individual)

Groups	*Count*	*Sum*	*Average*	*Variance*		
5(yes)	59	2500	42.37288	281.4103		
0(yes)	60	2733	45.55	250.15		
10(no)	36	2078	57.72222	236.0349		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	5523.77	2	2761.885	10.67073	4.61E-05	3.055558
Within Groups	39341.87	152	258.8281			
Total	44865.64	154				

A p-value of 4.61E-05 was found indicating that there is a significant difference in the average unique requirements addressed between at least two of the three groups. Further, comparing the average values between the three groups, 5(yes) has least average unique requirements addressed while 10(no) has the most average unique requirements addressed. This is similar to the pattern observed for addressing given requirements where 5(yes) group had least average given requirements addressed.

In order to investigate the average for which of the two pairs differed significantly, ANOVA was done between pairs of groups. Three possible pairs were investigated, 1) 5(yes) vs. 0(yes), 2) 0(yes) vs. 10(no) and 3) 10(no) vs. 5(yes).

Table 5-25 illustrates ANOVA results for comparing unique requirements addressed between 5(yes) and 0(yes) groups considering each solution as data point.

Table 5-25 ANOVA for 5(yes) vs. 0(yes) for unique requirements addressed (Individual)

Groups	*Count*	*Sum*	*Average*	*Variance*		
5(yes)-(Unique)	59	2500	42.37288	281.4103		
0(yes)-(Unique)	60	2733	45.55	250.15		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	300.2778	1	300.2778	1.130366	0.289887	3.922173
Within Groups	31080.65	117	265.6466			
Total	31380.92	118				

A p-value of 0.289887 indicates that there is no significant difference in average unique requirements addressed between 5(yes) and 0(yes) groups. While the students of both the groups were tasked with eliciting requirements, students of 5 (yes) groups were given a list of five requirements in addition to the problem statement. However, they perform comparably in terms of addressing the unique requirements indicating that there are no potential benefits of providing a list of requirements to students while also asking them to elicit more requirements.

Table 5-26 illustrates ANOVA results for comparing average number of unique requirements addressed between 0(yes) and 10(no) group considering each solution as data point.

Table 5-26 ANOVA for 0(yes) vs. 10(no) for unique requirements addressed (Individual)

Groups	*Count*	*Sum*	*Average*	*Variance*		
0(yes)-(Unique)	60	2733	45.55	250.15		
10(no)-(Unique)	36	2078	57.72222	236.0349		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	3333.667	1	3333.667	13.61267	0.000376	3.942303
Within Groups	23020.07	94	244.8944			
Total	26353.74	95				

A p-value of 0.000376 indicates that there is a significant difference in the average unique requirements addressed between 0(yes) and 10(no) group. While the difference between the two groups was not significant when comparing for given requirements, comparing the averages between the two groups, 10(no) group has greater average unique requirements addressed compared to 0(yes) group.

Table 5-27 shows ANOVA results for comparing average unique requirements addressed between 10(no) and 5(yes) groups considering each sketch as data point.

A p-value of 2.26E-05 indicates that there is significant difference in the average unique requirements addressed between two groups. Further comparing the average values, 10(no) has greater average unique requirements addressed compared to 5(yes) group. This is similar to the observation for comparing for given requirements where 0(no) group had greater average given requirements addressed compared to 5(yes) group. Again, this indicates that allowing the students to elicit their own requirements has no potential benefits in terms of more requirements addressed in the final solution.

Table 5-27 ANOVA for 10(no) vs. 5(yes) for unique requirements addressed (Individual)

Groups	*Count*	*Sum*	*Average*	*Variance*		
10(no)-(Unique)	36	2078	57.72222	236.0349		
5(yes)-(Unique)	59	2500	42.37288	281.4103		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	5267.571	1	5267.571	19.92774	2.26E-05	3.943409
Within Groups	24583.02	93	264.3335			
Total	29850.59	94				

From Table 5-25, Table 5-26, and Table 5-27, it is evident that 10(no) group performed better while considering the average number of unique requirements addressed in the solution. It was hypothesized that the students who were tasked with eliciting the requirements will be able to address more requirements in the solution. This stems from the fact that eliciting more requirements will allow the students to explore and comprehend the problem in a better way compared to the students who did not elicit any requirements. This will in turn result to more requirements addressed in the solution.

However, the findings suggest the contrary. The students who were tasked with elicitation of requirements actually perform poorly in terms of addressing the requirements. While not explicitly measured in this study, it is possible that the students who had the list of requirements were more confident about the requirements while the students who elicited the requirements were not very confident as their requirements were not externally validated. Thus, the students did not know whether the requirements they elicited were in fact 'true' requirements for the problem. This is reflected as fewer unique requirements are addressed by the students of 0(yes) group.

Next, section 5.8.4 discusses the findings from comparing unique requirements addressed between the three groups while considering each student as a data point.

5.8.4 Findings from comparing unique requirements (Integrated)

Table 5-28 illustrates ANOVA table for comparing average unique requirements addressed between the three groups considering each student as a data point.

Table 5-28 ANOVA for comparing for unique requirements (Integrated)

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
5(yes)-INT-UNI	16	1189	74.3125	182.6292		
0(yes)-INT-UNI	15	1102	73.46667	120.6952		
10(no)-INT-UNI	14	1061	75.78571	112.6429		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	39.71647	2	19.85823	0.141519	0.868451	3.219942
Within Groups	5893.528	42	140.3221			
Total	5933.244	44				

A p-value of 0.868451 was observed indicating that there is no significant difference in the average unique requirements addressed between the three groups when considering student as a data point.

While there was significant difference in the average number of unique requirements addressed between the three groups, there is no difference while considering each student as a data point. This shows that on average, each student is able to address same number of unique requirement irrespective of whether they are eliciting the requirements or not. Thus, while considering each student as a data point, the act of eliciting their own requirements does not seem to have any effect on the average unique requirements addressed.

Next, the fixation while addressing the requirements is investigated and the findings are discussed in section 5.9.

5.9 Findings for fixation while addressing requirements (RQ3.3)

RQ3.3 aims at investigating whether or not providing a list of requirements to the students results to fixation while addressing more requirements. It may be noted that the students of 0(yes) group were give only problem statement, the students of 5(yes) group were given problem and five requirements while the students of 10(no) group were given problem and list of 10 requirements. Thus, for the purpose of this analysis, the 'given' requirements refers to five, ten and fifteen requirements for 0(yes), 5(yes) and 10(no) group respectively.

Section 5.9.1 discusses the findings for fixation while considering each solution as a data point while section 5.9.2 discusses the findings for fixation while considering each student as a data point.

5.9.1 Comparing for fixation (Individual)

Table 5-29 illustrates the ANOVA results from comparing the ratio number of requirements addressed to number of requirements given to each group while considering each solution as a data point.

Table 5-29 Comparing average ratio of number of addressed/number of given for all groups (Individual)

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
5(yes) -Add/given	59	27.6	0.467797	0.030842		
0(yes)-Add/Given	60	36.4	0.606667	0.043345		
10(no)-Add/Given	36	19.6	0.544444	0.016127		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.5749	2	0.28745	8.89758	0.000222	3.055558
Within Groups	4.910591	152	0.032307			
Total	5.485491	154				

A p-value of 0.000222 indicates that there is significant difference in the average ratio of number of requirements addressed to the number of requirements given between at least two of the three groups. Further, comparing the average values, it is found that 0(yes) group had the maximum average ratio of number of requirements addressed to the number of requirements given, while the 5(yes) group had minimum average ratio of number of requirements addressed to the number of requirements given.

In order to further investigate which of the two pairs from the group differ significantly, an ANOVA was conducted between three possible pairs from the above groups. These are – 1) 5(yes) vs. 0(yes), 2) 0(yes) vs. 10(no) and 3) 10(no) vs. 5(yes).

Table 5-30 shows the ANOVA results from comparing 5(yes) and 0(yes) group for the average ratio of number of requirements addressed to number of requirements given while considering each solution as a data point.

Table 5-30 Comparing 5(yes) and 0(yes) for average ratio of number of addressed/number of given (Individual)

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
5(yes)-Add/given	59	27.6	0.467797	0.030842		
0(yes)-Add/Given	60	36.4	0.606667	0.043345		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.573685	1	0.573685	15.44383	0.000144	3.922173
Within Groups	4.346147	117	0.037147			
Total	4.919832	118				

A p-value of 0.000144 indicates that there is significant difference in the average ratio of number of requirements addressed to the number of requirements elicited between 5(yes) and 0(yes) groups. Further, comparing the average values, 0(yes) group has higher average ratio of number of requirements addressed to the number of requirements given compared to 5(yes) group.

Table 5-31 illustrates the ANOVA for comparing the average ratio of number of requirements addressed to the number of requirements given between 0(yes) and 10(no) group while considering each solution as a data point.

A p-value of 0.108675 indicates that there is no difference in the average ratio of number of requirements addressed to number of requirements given between the 0(yes) and 10(no) groups.

Table 5-31 Comparing 0(yes) and 10(no) for average ratio of number of addressed/number of given (Individual)

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
0(yes)-Add/Given	60	36.4	0.606667	0.043345		
10(no)-Add/Given	36	19.6	0.544444	0.016127		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.087111	1	0.087111	2.623007	0.108675	3.942303
Within Groups	3.121778	94	0.03321			
Total	3.208889	95				

Table 5-32 illustrates the ANOVA results for comparing the average ratio of number of requirements addressed to the number of requirements given between the 10(no) and 5(yes) group while considering each solution as a data point.

Table 5-32 Comparing 10(no) and 5(yes) for average ratio of number of addressed/number of given (Individual)

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
10(no)-Add/Given	36	19.6	0.544444	0.016127		
5(yes)-Add/given	59	27.6	0.467797	0.030842		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.13135	1	0.13135	5.190917	0.024995	3.943409
Within Groups	2.353258	93	0.025304			
Total	2.484608	94				

A p-value of 0.024995 indicates that there is significant difference in the average ratio of number of requirements addressed to number of requirements given between 10(no) and 5(yes) group. Further, comparing the average values, it can be observed that

10(no) group has higher average ratio of number of requirements addressed to the number of requirements given compared to 5(yes) group.

Thus, the results from Table 5-30, Table 5-31 and Table 5-32 indicate that while considering each solution as a data point, the students of 0(yes) group have the maximum ratio of number of requirements addressed to the number of requirements given. Thus, they perform better compared to the other two groups in terms of addressing the requirements given to them (for 0(yes), these are the requirements embedded in the problem statement). However, when comparing the average number of unique requirements addressed while considering each solution as a data point, 0(yes) group performs poorly (average-45.5) when compared to the 10(no) group (average-57.7). Thus, 0(yes) group seems to be fixated on addressing the 'given' requirements. This is indicated by the fewer unique requirements addressed by the students of 0(yes) group. This could stem from the fact that the students are not as much confident about the requirements elicited by them as they are about the 'given' requirements. Thus, they focus more on addressing the given requirements.

Further, 5(yes) group performs poorly compared to the other two groups while addressing both the given requirements and unique requirements. Thus, providing a list of requirements while asking the students to elicit the requirements does not seem to have any benefit while addressing requirements and seems to negatively affect the requirements addressed.

Next, section 5.9.2 provides a discussion of the findings from comparing the average ratio of number of requirements addressed to the number of requirements given for all three groups while considering each student as a data point.

5.9.2 Comparing for fixation (Integrated)

Table 5-33 illustrates the ANOVA results from comparing the three groups for average ratio of number of requirements addressed to number of requirements given while considering each student as a data point.

Table 5-33 Comparing average ratio of number of addressed/given for all groups (Integrated)

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
5(yes)-Add/given-INT	16	11.9	0.74375	0.011958		
0(yes)-Add/Given-INT	15	13.4	0.893333	0.039238		
10(no)-Add/Given-INT	14	9.866667	0.704762	0.010061		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.292844	2	0.146422	7.154978	0.002118	3.219942
Within Groups	0.859502	42	0.020464			
Total	1.152346	44				

A p-value of 0.002118 is observed indicating that there is statistical difference in the average ratio of number of requirements addressed to number of requirements given between at least two of the groups while considering each student as a data point.

In order to further investigate which of the two pairs from the group differ significantly, an ANOVA was conducted between three possible pairs from the above groups. These are – 1) 5(yes) vs. 0(yes), 2) 0(yes) vs. 10(no) and 3) 10(no) vs. 5(yes).

First, Table 5-34 shows the ANOVA results from comparing 5(yes) and 0(yes) groups for average ratio of number of requirements addressed to number of requirements given while considering each student as a data point.

Table 5-34 Comparing 5(yes) and 0(yes) for average ratio of number of addressed/number of given (Integrated)

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
5(yes)-Add/given-INT	16	11.9	0.74375	0.011958		
0(yes)-Add/Given)-INT	15	13.4	0.893333	0.039238		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.173227	1	0.173227	6.893825	0.013664	4.182964
Within Groups	0.728708	29	0.025128			
Total	0.901935	30				

A p-value of 0.013664 leads to the inference that there is significant difference in the average ratio of number of requirements addressed to the number of requirements given between 5(yes) and 0(yes) group. Further, comparing the averages, 0(yes) has a higher average ratio of number of requirements addressed to the number of requirements given compared to 5(yes) group.

Then, Table 5-35 illustrates ANOVA results from comparing 0(yes) and 10(no) group for average ratio of number of requirements addressed to number of requirements given while considering each student as a data point.

A p-value of 0.003524 leads to the inference that there is a significant difference in the average ratio of the number of requirements addressed to the number of requirements given between 0(yes) and 10(no) group while considering each student as a data point. Further, comparing the average values, 0(yes) group has a higher average ratio of number of requirements addressed to number of requirements given compared to 10(no) group.

Table 5-35 Comparing 0(yes) and 10(no) for average ratio of number of addressed/number of given (Integrated)

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
0(yes)-Add/Given-INT	15	13.4	0.893333	0.039238		
10(no)-Add/Given-INT	14	9.866667	0.704762	0.010061		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.257498	1	0.257498	10.22226	0.003524	4.210008
Within Groups	0.680127	27	0.02519			
Total	0.937625	28				

Finally, Table 5-36 illustrates ANOVA results for comparing 10(no) group and 5(yes) group for average ratio of the number of requirements addressed to the number of requirements given while considering each student as a data point.

Table 5-36 Comparing 10(no) and 5(yes) for average ratio of number of addressed/number of given (Integrated)

SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
10(no)-Add/Given-INT	14	9.866667	0.704762	0.010061		
5(yes)-Add/given-INT	16	11.9	0.74375	0.011958		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.01135	1	0.01135	1.024592	0.3201	4.195972
Within Groups	0.310169	28	0.011077			
Total	0.321519	29				

A p-value of 0.3201 indicates that there is no significant difference in the average ratio of number of requirements addressed to the number of requirements given between the 10(no) and 5(yes) group.

Thus, the results from Table 5-34, Table 5-35 and Table 5-36, indicate that while considering each student as a data point, 0(yes) has highest average ratio of the number of requirements addressed to number of requirements given and this is significantly higher compared to the other two groups. The students of 5(yes) and 10(no) group perform comparably as indicated by no significant difference in the average ratio of the number of requirements addressed to number of requirements given.

Further, comparing the average number of unique requirements addressed while considering each student as a data point, the students of all three groups perform comparably as there is no significant difference between the averages. However, 0(yes) group is able to address more given requirements compared to other two groups. This is similar to the observation made while comparing the average ratio of the number of requirements addressed to given considering each solution as a data point. This indicates that the students of 0(yes) group are fixated in addressing the given requirements.

The summary of findings from RQ3 is discussed in section 5.10.

5.10 Summary of Findings for RQ3-What impact do requirements have on outcomes?

RQ3 aimed at investigating specific roles that requirements can play in idea generation. To that end, a designer study was conducted with mechanical engineering students at Clemson University. As a part of the study, the students were divided into three groups. Each group was given the same design problem and was asked to develop design solutions for the given problem. The variable here was the number of requirements given to the students of the three groups. One group was given only problem statement [0(yes)], the second group was given problem statement and list of five requirements [5(yes)] and the third group was given problem and list of ten requirements [10(no)]. The students of 5(yes) and 0(yes) group also elicited requirements in addition to developing solutions for the given problem. Thus, the data collection entailed the elicited requirements and solutions developed by the students.

Further, three aspects were investigated. First, the type of requirements elicited and addressed by the students was investigated. Then the number of requirements addressed by the students was compared between the three groups and finally, the fixation caused while addressing the requirements was investigated.

The findings from comparing for the type of requirements elicited by the students reveal that for both 5(yes) and 0(yes) groups, more non-functional requirements were elicited compared to functional requirements. However, comparing the requirements addressed by the students it was found that the ratio of average number of non-functional and functional requirements addressed decreased. Thus, although the students elicit more non-functional requirements, their performance is comparable while addressing the requirements.

Next, the average number of given and unique requirements addressed in the solutions was compared between the three groups. It was hypothesized the act of eliciting the requirements will act as a primer and thus the groups that elicited requirements will have addressed more requirements in their solutions. However, the findings suggest that the group that was not tasked with eliciting requirements performed better than the other two groups that elicited requirements in terms of addressing the requirements.

Finally, the fixation in the requirements addressed was compared between the three groups. The findings suggest that 0(yes) group has higher average ratio of number of requirements addressed to number of given requirements compared to other two groups. Thus, 0 (yes) group performs better compared to the other two groups while addressing the given requirements. However, comparing the number of unique requirements, the 10(no) group performs better compared to the other two groups. Thus, the act of eliciting requirements results to fixation while addressing the requirements.

CHAPTER SIX CONCLUSIONS AND FUTURE WORK

The overall goal of this dissertation is to develop a systematic understanding of the current use of requirements in conceptual design. To achieve this overall goal, three broad questions were investigated through the research presented in this thesis:

- RQ1 Requirements-What are we teaching?
- RQ2 How are students using requirements? and
- RQ3 What impact do requirements have on outcomes?

First, the critical gaps in requirements education were explored through a survey of design text books and survey of faculty. While the survey of faculty reveals that the requirements related activities are performed throughout the design project, most tools mentioning the use of requirements are found only in the conceptual phase as evident from the survey of design textbooks. Thus, a significant gap is identified in terms of lack of sufficient tools explaining the use of requirements throughout the design process.

The consequences of the lack of sufficient tools explaining the use of requirements throughout the design process are further realized while exploring how students are using requirements (RQ2). There is difference in the number of requirements elicited by the student teams working on the same project. While this difference is expected in the final week, it is expected that students have similar number of requirements in initial week as they have the same input in terms of the requirements established by the sponsor. Thus, the different numbers of elicited requirements in initial week indicate that the students are following different elicitation practices. Further, this could also stem from the fact that there are not many tools that allow the students to systematically elicit requirements. The completeness and specificity of requirements increases from initial week as expected however too specific requirements are the ones that are not fulfilled in the final design solution.

Another finding from investigating the student requirements is that the requirements document of novice designers changes and the change occurs in multiple ways. Currently, the students are not mandated to maintain weekly updates on requirements. Further, few tools mentioned in the design textbooks allow the students to systematically maintain and trace the changes within requirements. This could lead to loss of important information pertaining to the project and ultimately lead to project failures. Thus, it is recommended that students should be taught appropriate tools to manage the project requirements and trace the changes. Perhaps, the weekly requirements update sheet can serve as a starting point to that end.

Finally, to identify what role requirements could potentially play in conceptual design, specifically idea generation, a designer study was conducted. First, analyzing the requirements elicited by the students, it is found that more non-functional requirements are elicited compared to the functional requirements. Same observation is made for both the groups tasked with eliciting requirements. Further, analyzing the solutions developed by the students of all three groups for requirements addressed, it is found that the students given the requirements performed better in terms of addressing the requirements compared to the two groups tasked with eliciting requirements. Thus allowing the students to elicit requirements had no benefit in terms of requirements addressed in the final solution.

Based on the findings from the studies conducted for the purpose of this thesis, several recommendations can be made. These recommendations will serve as guidelines for improving the requirements education and application of requirements education. These recommendations and their potential benefits are discussed below:

- **Recommendation 1**: It is recommended that the students should be taught the concepts of requirements completeness.
- **Benefit:** While the students have preliminary idea of requirements elicitation, they are not formally taught to judge whether or not the elicited requirements are 'good' requirements. It is evident from the literature that one of the desirable characteristics of 'good' requirements is that they should be complete [20]. From the case study, it was found that some teams had more completeness in the initial weeks than others. This inconsistency suggests that completeness as a concept has not been fully digested by the students. Thus, teaching the concept of completeness of requirements will allow the students to identify poorly defined requirements and thus introduce them to the practice of writing 'good' requirements.
- **Recommendation 2:** It is recommended that the students should be taught the concepts of requirements specificity.
- **Benefit:** Specificity is essentially a measure of level of detail within a requirement as a count of adjuncts or numerical values. It is desirable that

requirements are non-ambiguous and testable [20]. By adding details to the requirements in terms of adjuncts, the ambiguity can be reduced. Further, the numerical values within requirements make them testable. Thus, teaching the concept of requirement specificity will allow the students to internally measure the 'goodness' of the requirements elicited by them. Further, it will also help the students to identify poorly defined requirements and add necessary details to it to reduce the ambiguities.

- **Recommendation 3:** It is recommended that the students should be taught tools or methods for managing the requirements document.
- **Benefit:** It is evident that the requirements documents of novice designers change in multiple ways. Currently, the students are not taught any tools that would allow them to manage the requirements document and trace the changes systematically. Thus, teaching the tools that would allow the students to manage the requirement document will help mitigate the negative effects of losing valuable information in form of requirements changes.
- **Recommendation 4:** It is recommended that the students should be required to submit weekly requirements update along with executive summary and presentations.
- **Benefit:** Mandating the submission of weekly requirements update has twofold benefits. First, it will allow the students to manage the requirements and systematically track the changes in the requirements throughout the design project. Additionally, it will allow the future researchers to collect valuable

information to study requirements evolution without intervening in the natural process followed by the students.

- **Recommendation 5**: It is recommended that the students should be encouraged to use design solutions to identify new requirements.
- **Benefit** The findings from the designer study reveal that the solutions generated by the students fulfill more requirements than those given to or elicited by individual students. This then means that the preliminary solutions can be used to identify new requirements. Thus, this will allow the students to use preliminary concept as 'means' for identifying requirements that they would otherwise miss.

6.1 Intellectual Merit

The intellectual merit of this research lies in developing a systematic understanding of how requirements are currently used in the conceptual design phase, and specifically for idea generation activities. The ultimate goal of this research is to develop guidelines and recommendations for use of requirements in idea generation. Prior to developing guidelines and recommendations, it is necessary to investigate the current state of requirements education so that critical gaps can be identified. This will be done through systematic design and application of surveys and case study, which will require a thorough study of these research methods. While the survey and case study will aid in identifying critical gaps in imparted and applied requirements education, specific investigation of requirements elicitation and usage in idea generation will be achieved by systematically designing and conducting a designer study. Design and implementation of user study will require identification of critical variables of interest and carefully controlling other factors to obtain valuable results.

To summarize, the intellectual merit of this research lies in the following research contributions:

1. Developing a systematic understanding of the current use of requirements in conceptual design
2. Systematic design and implementation of research methods such as surveys, case study and designer study to answer the research questions.

6.2 Broader Impact

As recognized in literature, a considerable amount of design process time is spent in requirements elicitation. However, clear guidelines on how to use the elicited requirements for generating ideas seem to be missing. This is also evident from surveying the mechanical engineering design textbooks. Of the wide variety of idea generation tools discussed in the design textbooks, most lack details and specifics on using requirements for idea generation. Specifically, the details, such as how many and what type of requirements may be used while generating design ideas for a given problem are missing. Requirements seem to be mostly used for concept evaluation rather than concept generation [72]. Novice designers, who rely heavily on classroom education and textbooks for learning how to use requirements at various stages of design process, are thus left poorly informed about what to do with requirements once they are elicited.

Thus, the broader impact of this research lies in beginning to provide systematic guidelines for instructors and designers on using requirements in idea generation. These guidelines may help to significantly reduce the ambiguities associated with using requirements while generating ideas. Instead of using requirements as just a means to validate developed concepts, using the guidelines may encourage designers to use requirements as an input for the conceptual design phase.

6.3 Future Research

The primary goal of this research is develop a systematic understanding of use of requirements within conceptual design. The findings from the tasks are used to make recommendations. However there are several limitations of this research as discussed below:

- First, only one project was investigated for understanding how are students using requirements. While this project is representative of a typical senior design project, extensive study spanning multiple projects is necessary to strengthen the confidence in the findings. Further, studying multiple different projects will allow mitigating the effects variables such as the nature of design project, composition of teams, composition of advisory committee and feedback received through design reviews.
- Due to the limitation of data collected, comparisons could only be made between the requirements documents generated in initial weeks and final week. Thus, this does not allow to completely capture the evolution of requirements throughout the project, rather it is a comparison between initial and final states. Thus, extensive

study exploring the requirements evolution in each week will provide a better picture of the changes in requirements.

- The designer study to investigate the role of requirement in idea generation was conducted with single pool of senior design students. While the results show interesting findings, future studies can be conducted in ME-401 and ME-402. This will aid in comparison to see if the knowledge gained during the pre-capstone class has any benefit in the results.
- Protocols developed for analyzing the solution was tested for inter-rater reliability (IRR). However, the IRR values were on the lower side. This means that there is scope for improvement. So more robust protocols can be developed and tested.
- The findings from the designer study reveal that the solutions can potentially be used to generate more requirements. While the exact benefits of using solutions to elicit more requirements are not currently investigated, future designer studies can be conducted to understand the iterative process of requirements-solutions-requirements. This will open new avenues for using requirements in conceptual design.

Thus, the future work of this research entails addressing the limitations of this research and further exploring new avenues for application of requirements in conceptual design.

BIBLIOGRAPHY

[1] P. G. Beitz W., Engineering Design: A systematic Approach, 3rd ed., London: Springer-Verlag London Limited, 2007.

[2] J. McLellan, B. Morkos, G. G. Mocko and J. D. Summers, "Requirements modeling systems for mechanical design: A Systematic Method for Evaluating Requirement Management Tools and Languages," in *ASME, International Design Engineering Technical Conference*, Montreal, Canada, 2010.

[3] B. Morkos, *Reasoning and Representation Support For Requirements Change Driven Design Management,* 2010.

[4] E. Hull, K. Jackson and J. Dick, Requirements Engineering, Springer - Verlag, 2005.

[5] J. Goguen and C. Linde, "Techniques of Requirements Elicitation," in *1st IEEE International Symposium on Requirements Engineering*, San Diego, 1993.

[6] A. Chakrabarti, S. Morgenstern and H. Knaab, "Identification and application of requirements and their impact on the design process: a protocol study," *Research in Engineering Design,* vol. 15, no. 1, pp. 22-39, 2005.

[7] E. Worinkeng, S. Joshi and J. D. Summers, "An Experimental Study: Analyzing Requirement Type Influence on Novelty and Variety of Generated Solutions," *International Journal of Design Creativity and Innovation,* Submitted 2013 (In Review).

[8] K. T. Ulrich and S. D. Eppinger, Product design and Development, McGraw-Hill, Inc, 1995.

[9] D. G. Ullman, T. . G. Dietterich and L. . A. Stauffer, "A Model of the Mechanical Design Process based on Empirical Data," *AIEDAM,* vol. 2, no. 1, pp. 33-52, 1988.

[10] K. Otto and K. Wood, Product Design- Techniques in Reverse Engineering and new product development, New Jersey: Prentice Hall, 2001.

[11] R. J. Eggert, Engineering Design, Meredian, Idaho: High Peak Press, 2010.

[12] G. Prudhomme, F. Pourroy and J. D. Summers, "Enriching Requirement-Activities in Design through French-US Instruction Comparison," in *International Conference on Engineering Design*, Seoul, Korea, 2013.

[13] L. Chung and B. Nixon, "Dealing with Non-functional Requirements: Three Experimental Studies of Process Oriented Approach," in *17th International Conference on Software Engineering*, Seattle, WA, 1995.

[14] P. Shankar, B. Morkos and J. D. Summers, "A Hierarchical Modeling Scheme with Non Functional Requirements," in *in ASME Design Engineering Technical Conference*, Montreal, Canada, 2010.

[15] H. Holbrook, "A Scenario-based methodology for conducting requirements elicitation," *SIGSOFT Software Engineering Notes,* vol. 15, no. 1, pp. 94-104, 1990.

[16] B. Morkos, "Computational Representation and Reasoning support for Requirements Change Management in complex system design-A PhD Thesis," Clemson University, Clemson, 2012.

[17] C. Lamar, "Linguistic Analysis of Natural Language Engineering Requirements -A MS thesis submitted to Graduate school of Clemson Universtiy," Clemson University, Clemson, 2009.

[18] I. Hooks, "Writing Good Requirements," in *Proceedings of Third International Symposium of the NCOSE*, 1993.

[19] W. Wilson, L. H. Rosenberg and L. E. Hyatt, "Automated Quality Analysis of Natural Language Requirements Specifications," in *In Proceedings of PNSQC Conference*, 1996.

[20] "IEEE Recommended Practice for Software Requirements Specification," IEEE Std-830 (Revision of IEEE Std 830-1993, 1998.

[21] A. Lash, "Computational Representation of Linguistics Semantics forRequirements Analysis in Engineering Design-MS Thesis," Clemson University, Clemson, 2013.

[22] W. Lam and V. Shankararaman, "Requirements Change: A Dissection of Management Issues," in *25th Euromicro Conference*, 1999.

[23] R. C. Sugden and M. R. Strens, "Strategies, tactics and Methods for Handling Change," in *IEEE Symposium and Workshop on Engineering of Computer-Based Systems*, 1996.

[24] S. Harker, K. Eason and J. Dobson, "The Change and Evolution of Requirements as a Challenge to the practice of Software Engineering," in *IEEE International Symposium on Requirements Engineering*, San Diego, California, 1993.

[25] P. Shankar, "Development of Design Method to reduce Change propogation - A

PHD Dissertation submitted to graduate school of Clemson University," Clemson University, Clemson, 2011.

[26] T. Ezhilan, "Modelling Requirements Propogation to Generate Solutions for minimizing Mass-MS Thesis," Clemson University, Clemson, 2007.

[27] J. M. McLellan, "A Proposed Method to Identify Requirements Significant to Mass Reduction-MS Thesis," Clemson University, Clemson, 2010.

[28] G. Gabrysiak, H. Giese and A. Seibel, "Why Should I help you to Teach Requirements Engineering?," in *Requirements Engineering Education and Training -6th International Workshop*, Trento, Italy, 2011.

[29] T. Nakatani, T. Tsumaki and T. Tamai, "Requirements Engineering Education for Senior Engineers:Course Design and its Evaluation," in *Requirements Engineering Education and Training - 5th International Workshop*, 2010.

[30] "Engineering Criteria 2000: Criteria for accrediting programs in engineering in the United States," Accreditation Board for Engineering and Technology, Baltimore, 2000.

[31] R. H. Todd, S. P. Magleby, C. D. Sorensen, B. R. Swan and D. K. Anthony, "A survey of Capstone Engineering Courses in North America," *Journal of Engineering Education,* no. April, pp. 165-175, 1995.

[32] S. Howe, "Where are we Now? Statistics on Capstone Courses Nationwide," *Advances in Engineering Education,* vol. 2, no. 1, pp. 1-27, 2010.

[33] B.-D. Ljerka and I. Alexander, "Learning how to discover requirements," in

Requirements Engineering Education and Training, 2008.

[34] C. J. Atman, K. Yasuhara, R. S. Adams, T. J. Barker, J. Turns and E. Rhone, "Breadth in Problem Scoping: a comparison of Freshman and Senior Engineering students," *International Journal of Engineering Education,* vol. 24, no. 2, pp. 234-245, 2008.

[35] N. Cross, Designerly ways of Knowing, London: Springer-Verlag, 2006.

[36] R. K. Yin, Case Study Research: Design and Methods, Thousand Oaks, California: SAGE Publications, 2003.

[37] "The Case Study as a Research Method - Uses and Users of Information LIS 391D.1," Spring 1997. [Online]. Available: http://www.gslis.utexas.edu/~ssoy/usesusers/l391d1b.htm.

[38] S. Teegavarapu and J. D. Summers, "Case Study Method for Design Research," in *Proceedings of IDETC/DTM*, New York , 2008.

[39] B. Morkos and J. D. Summers, "Requirements Change Propogation Prediction Approach: Results from an Industry Case Study," in *Proceedings of ASME International Design Engineering Technical Conference*, Monreal, Canada, 2010.

[40] D. Stowe, "Investigating the Role of Prototyping in Mechanical Design Using Case Study Validation- MS Thesis," Clemson University, Clemson, 2008.

[41] D. Veisz, E. Z. Namouz, S. Joshi and J. D. Summers, "CAD vs. Sketching: An Exploratory Case Study," *CAD-AIEDAM Special Issue,* vol. 26, no. 3, pp. 1-46, 2012.

[42] B. Morkos and J. D. Summers, " Implementing Design Tools in Capstone Design Projects: Requirements Elicitation Through Use of Personas," in *Capstone Design Conference*, Boulder, Colorado, 2010.

[43] S. Joshi and J. D. Summers, "Investigating Information Loss in Collaborative Design: A Case Study with Capstone Design Project," in *Capstone Design Conference*, Boulder, Colorado, 2010.

[44] S. Joshi, B. Morkos and J. D. Summers, "Requirements Analysis: Case Study with Capstone Design Project," in *Capstone Design Conference*, Champaign, 2012.

[45] S. Teegavarapu, S. Miller and J. D. Summers, "On the use of design methods by Capstone design Students," in *Proceedings of ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, NewYork, 2008.

[46] Z. Y. Chen, Y. Shengji, J. Q. Lin, Y. Zeng and A. Eberlein, "Formalization of Product Requirements: from natural language descriptions to formal specifications," *International Journal of Manufacturing Research,* vol. 2, no. 3, pp. 362-387, 2007.

[47] M. Lehman and L. Belady, Program evolution: Process of Software Change, Academic Press, 1985.

[48] G. Auriol, C. Baron and J.-Y. Fourniols, "Teaching requirements skulls within the context of a physical engineering project," in *Requirements Engineering Educational Training*, 2008.

[49] D. Callele and D. Makaroff, "Teaching Requirements Engineering to an

unsuspecting audience," in *Proceedings of the 37th SIGCSE technical symposium on Computer Science Education*, New York, NY, 2006.

[50] O. Gotel, V. Kulkarni, M. Say, C. Scharff and T. Sunetnanta, "Distributing Responsibilities to Engineering Better Requirements :Leveraging Knowledge and Perspectives for students to learn a key skill," in *Requirements in Engineering Education and Training*, 2008.

[51] C. Dym and P. Little, Engineering Design: A Project Based Introduction, NewYork, NY: John Wiley, 2000.

[52] S. Pugh, Total Design: Integrated Methods for successful product engineering, Workingham, England: Addison-Wesley Pub.Co, 1991.

[53] G. Deiter, Engineering Design: A Materials and Processing Approach, New York: McGraw Hill, 2000.

[54] N. Cross, Engineering Design Method-Strategies for product design, West Sussex, England: John Wiley & Sons, 2010.

[55] T. Woodson, Introduction to Engineering Design, New York, USA: McGraw Hill, 1966.

[56] B. Hyman, Fundamentals of Engineering Design, New Jersey, USA: Prentice Hall Inc., 1998.

[57] M. Glinz, "Rethinking the Notion of Non-functinal Requirements," in *Proceedings of the Third World Congress for Software Quality*, 2005.

[58] I. Sommerville, Software Engineering, Addison Wesley: Pearson, 2004.

[59] C. W. Hoover and J. B. Jones, Improving Engineering Design: Designing for Competitive Advantage, Washington, DC: National Research Council, National Academy Press, 1991.

[60] P. A. Glasow, "Fundamentals of Survey Research Methodology," MITRE, Washington C3 Center, McLean, Virginia, 2005.

[61] S. Howe and J. Wilbarger, "2005 National Survey of Engineering Capstone Design courses," in *ASEE Annual Conference and Exposition*, 2006.

[62] J. Wilbarger and S. Howe, "Current Practices in Engineering Capstone Education:Further Results from a 2005 Nationwide Survey," in *36th ASEE/IEEE Frontiers in Education Conference*, San Diego, CA, 2006.

[63] W. Tellis, "Introduction to Case study," *The Qualitative Report (Online Journal),* vol. 3, no. 2, pp. 1-12, 1997.

[64] S. Joshi, "Mapping Problem and Requirements to Final Solution: Document analysis of senior design projects," MS Thesis , Clemson, 2010.

[65] B. Morkos, S. Joshi and J. D. Summers, "DETC2012-71417 Representation: Formal Development and COmputational Recognition of Localized Requirement Change Types," in *Proceedings of ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Chicago, 2012.

[66] C. Rolland, C. Salinesi and A. Etien, "Eliting Gaps in Requirements Change," *Requirements Engineering,* vol. 9, pp. 1-15, 2004.

[67] "ISO 11442-6," 1996.

[68] G. E. Stark, P. Oman, A. Skillicorn and A. Ameele, "An examination of the effects of requirements change on software maintenance releases," *Journal of Software Maintenance : Research and Practice,* vol. 11, no. 5, pp. 293-309, 1999.

[69] L. Lin and J. H. Poore, "Pushing Requirements Changes through to Changes in Specifications," *Frontiers of Computer Science in China,* vol. 2, no. 4, pp. 331-343, 2008.

[70] M. Chaudron, "Requirements in Engineering -Presentation," 2007.

[71] Z. Jiayi, L. Yunjuan and G. Yuesheng, "The Requirements Change Analysis for Different level users," in *International Symposium on Intelligent Information Technology Application Workshops*, Shanghai, 2008.

[72] S. Joshi, B. Morkos, P. Shankar, J. D. Summers and G. M. Mocko, "Requirements in Engineering Design: What are we teaching?," in *Proceedings of the Tools and Methods of Competitive Engineering 2012*, Karlsruhe, Germany, 2012.

[73] P. Shankar, B. Morkos and J. D. Summers, "Reasons for change propagation: a case study in an automotive OEM," *Research in Engineering Design,* vol. 23, no. 4, pp. 291-303, 2012.

[74] B. Morkos, P. Shankar and J. D. Summers, "Predicting Requirement Change Propagation Using Higher Order Design Structure Matrices: An Industry Case Study," *Journal of Engineering Design,* vol. 23, no. 12, pp. 905-926, 2012.

[75] "Qualitative Research Methods: A data collector's field guide - Module 2 Participant

Observation," Family Health Internatinal.

[76] L. C. Schmidt, N. V. Hernandez, G. Kremer and J. Lindsey, "Pilot of Systematic Ideation Study with Lessons Learned," in *Proceedings of ASME Internation Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Montreal, Canada, 2010.

[77] J. Linsey, M. G. Green, J. T. Murphy, K. L. Wood and A. B. Markman, ""Collaborating to Success": An Experimental Study of Group Idea Generation Techniques," in *Proceedings of ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Long Beach, California, 2005.

[78] J. S. Linsey, E. F. Clauss, T. Kurtoglu, J. T. Murphy, K. L. Wood and A. B. Markman, "An Experimental Study of Group Idea Generation Techniques: Understanding the roles of Idea Representation and Viewing Methods," *Journal of Mechanical Design,* vol. 133, pp. 1-15, March 2011.

[79] C. Sen, "A Formal Representation of Mechanical Functions to support Physics-based Computational Reasoning in Early Mechanical design-A Dissertation presented to Graduate School of Clemson University," Graduate School of Clemson University, Clemson, SC, 2011.

[80] R. Ramchandran, "Understanding the role of functions and interactions in the Product Design," Clemson, 2011.

[81] R. Hannah, S. Joshi and J. D. Summers, "A user study of interpretability of

engineering design representations," *Journal of Engineering Design,* vol. 23, no. 6, pp. 443-468, 2012.

[82] R. Hannah, "User Study of Information Extracted from Engineering Representations -MS thesis," Clemson University, Clemson, 2009.

[83] J. J. Shah, "Experimental Investigation of Progressive Idea Generation Techniques in Engineering Design," in *Proceedings of ASME International Design Engineering Technical Conference*, Atlanta, GA, 1998.

[84] T. Kurtoglu, M. I. Campbell and J. S. Linsey, "An experimental study on the effects of a computational design tool on concept generation," *Design Studies,* vol. 30, no. 6, pp. 676-703, 2009.

[85] A. Dong, A. W. Hill and A. M. Agogino, "A Document analysis method for characterizing design team performance," *Journal of Mechanical Design,* vol. 126, pp. 378-385, May 2004.

[86] A. Dong, "Quantifying coherent thinking in design: A computational linguistic approach," *Design Computing and Cognition,* pp. 521-540, 2004.

[87] S. Song, A. Dong and A. M. Agogino, "Time cariation of design "Story telling" in engineering design teams," in *International Conference on engineering design*, Stockholm, 2003.

[88] K. Miller and M. Vanni, "Inter-rater Agreement Measures and the Refinement of Metrics in the PLATO," in *In Proceedings of MT Summit X*, Phuket, Thailand, 2005.

[89] B. Morkos, "Computational Representation and Reasoning Support for

Requirements Change Management in Complex System Design," Clemson, SC, 2012.

APPENDICES

Appendix A:Survey of Faculty

Part A – Course Information Questions

1. Institution - ______________________________ Department -___________________________________

2. How long has this course been in existence in its present form? ____________ Years

3. What prerequisites exist for the senior capstone design course? List the course name

4. In what type of senior capstone design course does your department participate?
   [ ] Individual students solve engineering design projects
   [ ] Departmental student teams solve engineering design projects
   [ ]Interdisciplinary or inter-departmental student teams solve engineering design projects
   [ ] Other (Please describe):

5. Which of the following best describes the capstone design course?
   [ ] Independent project without formal class instruction
   [ ] Formal class instruction followed by students independently completing the project
   [ ] Formal class carried out in parallel with a major design project
   [ ] Formal class only using smaller design assignments for application purposes
   [ ] Others (Please describe):

6. What is the duration of the class instruction and design project?

Class		**Project**	
	[ ] 1 Quarter		[ ] 1 Quarter
	[ ] 1 Semester		[ ] 1 Semester
	[ ] 2 Quarters		[ ] 2 Quarters
	[ ] 2 Semesters		[ ] 2 Semesters
	[ ] Other (Please specify):		[ ] Other

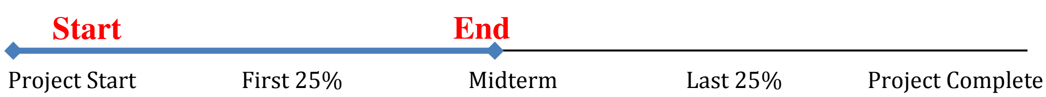
7. Which of the following are used to aid the instruction for capstone design course? (check all that apply)
   [ ] Textbook (Specify the name):
   [ ] Instructor notes
   [ ] Class Website
   [ ] Workshops
   [ ] Other (Please specify):

8. If the project have formal class instruction, how many hours of instructions do students receive each week?
   [ ] 1 [ ] 4

[ ] 2 [ ] 5
[ ] 3 [ ] 6+
[ ] Other (Please explain):

Part B - Requirements Usage Information

Given below are some requirements related activities. Here “requirements” refer to both project (requirements for the design given by sponsor and/or elicited by students) and process (requirements on time and resources) requirements. The horizontal scale below each activity is indicative of the total time period from start of the project to the end of the project. On this scale, for each activity, mark ‘start’ and ‘end’ from the time when you would expect the students to start and end that particular activity during the design project. This marking is illustrated below:



If a particular activity is not performed by the students, do not mark anything on the scale and write N/A in front of the activity.

a. **Requirements elicitation – gathering requirements at the start of the project**

Project Start First 25% Midterm Last 25% Project Complete

b. **Requirements documentation – documenting the requirements that are elicited**

Project Start First 25% Midterm Last 25% Project Complete

c. **Requirements verification from the customer – checking with customer to ensure that correct requirements are elicited**

Project Start First 25% Midterm Last 25% Project Complete

d. **Using requirements for concept generation – generating concepts by using the elicited requirements**

Project Start	First 25%	Midterm	Last 25%	Project Complete

e. **Using requirements for concept evaluation – using requirements to evaluate concepts generated in conceptual design phase and eliminating the concepts that do not meet requirements**

Project Start	First 25%	Midterm	Last 25%	Project Complete

f. **Requirement update/change – adding, deleting or modifying the requirements from existing requirements list**

Project Start	First 25%	Midterm	Last 25%	Project Complete

g. **Solution validation – ensuring that the final solution meets all requirements**

Project Start	First 25%	Midterm	Last 25%	Project Complete

Part C – Tools for Requirements

Given below is a list of design tools. Answer the questions in column 2 and 3 in 'yes' or 'no'.

Design tool	Is the design tool formally taught in capstone or other prerequisite design courses?	Are students using this design tool as evident from design reports or other written documentation such as weekly summary, presentation or design notebooks?
Requirements checklist		
Project definition sheet		
Quality function deployment		
Objective tree		
Morph chart		
Function means tree		
Decision matrix		
Pair wise comparison		
Requirements validation checklist		
OTHER		
OTHER		
OTHER		

Appendix B: Example Problems from Literature

Design Problem	Characteristics					
	Within domain knowledge	Unfamiliar	Neutral representati on	Motivating for students	Solvable in stipulated time	Possibility of many solutions
"A portable human-powered device is required which will extract fence posts in remote areas. The fence posts are made of wood of square cross-section with sides of between 2 and 4 inches, are between 6 and 7.5 feet long, and may have been sunk up to 3 feet into the ground. An initial vertical force of up to 500 lb. may be required to extract the posts, which must be in a reusable condition afterwards." [83]	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
"Design a device that takes water, sodium bicarbonate (gas), and soda flavor syrup as input and mixes them into a soda drink. The device is targeted as a home type kitchen appliance. The inputting of the water can be accomplished through a standard kitchen faucet. Please assume that the soda flavor syrup is available in a separate container that can be poured into the device you are designing, and the sodium bicarbonate is contained in a canister that can safely transfer sodium bicarbonate into the system." [84]	**Yes**	**No**	**Yes**	**Yes**	**Yes**	**Yes**
"Bottle Capping Device: 'Design a machine that registers a bottle to a capping station, caps it, and allows somebody to retrieve the capped bottle from the device. Please do not limit your design to a particular bottle, cap geometry. You can assume that you have control over the specifics of both these system inputs and how they should interface with each other'" [84]	**Yes**	**No**	**Yes**	**Yes**	**Yes**	**Yes**
"Mr. Smith is hosting a party next month and has invited his colleagues. He wants a new design for a home use burrito-folding machine. The device must adhere to the following design requirements: *1. Deliver completed burritos at a rate of at least 4 burritos per minute* *2. The device must fit on a counter top* *3. Position empty tortilla to store fillings* *4. Fill the tortilla after proper positioning* *5. Wrap burrito over the filling* *6. Easy to install* *7. Easy to use* *8. The device must be easy to clean after use (<15 minutes and no special tools)* *9. The device must be safe – cause no injury to the user."* [80]	**Yes**	**Yes**	**No**	**Yes**	**Yes**	**Yes**

Appendix C: Exit Survey for Designer Study

1- Provide your initials here ……………

2- How confident were you about understanding the problem?

	1	2	3	4	5	
Not confident	( )	( )	( )	( )	( )	Very confident

3- Were the given requirements sufficient for you to generate solutions?

	1	2	3	4	5	
Not at all sufficient	( )	( )	( )	( )	( )	Very sufficient

4- Did you use all the requirements while designing concepts?

Yes………….. No ……………..

5- What requirements did you explicitly address in your concepts? List the requirements in the space below.

6- List the requirements that were easy to address while sketching concepts.

7- List the requirements that were difficult to address while sketching concepts.

8- Did you develop new requirements while sketching concepts?

Yes…………..   No ……………..

9- List the requirements that you developed while sketching concepts.

10- While sketching, how strong was your focus on developing as many concepts as possible?

	1	2	3	4	5	
Not at all	( )	( )	( )	( )	( )	Very Strong

11- While sketching, how strong was your focus on developing concepts that were new?

	1	2	3	4	5	
Not at all	( )	( )	( )	( )	( )	Very strong

12- While sketching, how strong was your focus on developing concepts that were of many different types?

	1	2	3	4	5	
Not at all	( )	( )	( )	( )	( )	Very strong

13- While sketching, how strong was your focus on developing concepts that were feasible?

	1	2	3	4	5	
Not at all	( )	( )	( )	( )	( )	Very strong

Appendix D: Ideation Sheet

Concept title:

Initials

1

Appendix E: Requirement Elicitation Sheet

Use the categories provided in the table below to guide the requirements elicitation. Note that the requirements you elicit do not necessarily have to fit in the given category. Use the categories only as a guideline. There is no limit to the number of requirements that you can elicit.

Categories			
Geometry	Material	Production	Operation
Kinematics	Signals	Quality control	Maintenance
Forces	Safety	Assembly	Recycling
Energy	Ergonomics	Transport	Cost

List your requirements in the box provided below, limiting to only one requirement per box:

Appendix F: Guidelines for Identifying Means

Code	Requirements (Embedded in problem statement or given to students)	Guidelines for identifying means
P1	The device must allow the disabled in wheel chair to experience the joy of picking peaches from the tree.	The solution must have indication of allowing user to pick the fruit from tree. For example a ramp that allows user to touch fruit
P2	The device must allow the disabled in wheel chair to collect peaches in basket while still in the wheel chair	The solution must indicate that the user does not require getting out of the chair in order to pick the fruit. For example extendable arms that allow to pick fruit,
P3	The device must be manually operated.	The solution must have indication of use by hand such as lever or gear system, must not have automation
P4	The device must prevent damage to the fruit while picking it.	The solution must not have sharp edges or grips that could damage the fruit.
P5	The fruit should not fall on the ground while picking	The solution must indicate a means that prevents the fruit from falling to the ground. This could some sort or guard or basket that prevents the fruit from falling
G1	The device must be able to reach heights in the range of 8-10 feet while used by a person in wheel chair	The solution must have a means that allows it to extend in the range of 8-10 feet such as extendable pole or scissor mechanism.
G2	The device must allow the user to pick multiple fruits at a time	The solution must indicate means to allow to pick multiple fruits . For example, shears allow to cut multiple fruits at a time
G3	The device must be able to grasp the fruit.	The solution must indicate means to grasp the fruit such as grips or jaws.
G4	The device must be able to hold the fruit until the fruit is put in the basket	The solution must indicate a means for holding the fruit from the time that it is picked till put in the basket. For example a small basket attached with cutter or a funnel device that transports fruit to the basket
G5	The device must provide an indication to the user when the fruit has been picked	The design must have some kind of visual signal or auditory signal or weight sensor that indicates fruit has been picked

G6	The forces required to operate the device must be within the upper body strength of a person in wheel chair	Indication that the user does not need a lot of effort to operate the design. For example use of mechanisms such as lever, push buttons
G7	The device must be safe to use	No sharp edges, no heavy elements, no overhang
G8	When not in use, the device should fit in 4 feet by 3 feet storage space	Mention of the size on the solution sheet, or mention a scale which represents the storage size
G9	The device must not be made of corrosive materials	Mention the material used for building the design
G10	The device must not cost more than $50	Indication such as very simple design, actual cost written on the design
Code	**Unique ALL requirements (elicited by students)**	**Guidelines for identifying means**
1	The design should be easy to assemble (ease of assembly)	The design must have less than 20 parts, features such as snap fit. The design is not easy to assemble if it has screws, nuts, bolts, special assembly requirements
2	Device should require minimal assembly	The design must have less than 20 parts
3	The design should be easy to disassemble	The design must have less than 20 parts, features such as snap fit. The design is not easy to disassemble if it has screws, nuts, bolts, special assembly requirements
4	(The device) must not need assembly of parts	Indication that no assembly is required, one part design, design comes as add-on wheel chair and does not require assembly
5	(The device) must be quick to assemble	The design must have less than 20 parts. The assembling process is simple and quick, for example if the design has many screws or bolts, it will take longer to assemble
6	(The device) must be assembled alone	The design must not have heavy parts, or require special tools to assemble
7	Assembly-There should be minimal parts to ensure ease of storage	Indication such as less than 20 parts, compact design. The storage size is mentioned on the solution sheet
8	Device must not have more than 4 components to be assembled	Solution must have less than 4 parts to assemble

9	Time to assemble the device must be less than 5 minutes	Assembly time written on the solution sheet
10	(The device) must cost less than $ 50	Indication such as very simple design, actual cost written on the design
11	(The device) must be affordable to a typical user	Cost of the design is written on the solution sheet
12	Ergonomics -The design should be painless	No pain causing elements such as heavy effort required to operate the design, uncomfortable posture to operate the design, no sharp edges
13	Ergonomics-The design should require minimal amount of energy for crippled people	Indication of minimal effort for user such as small lever, trigger, string pull, push button
14	(The controls must) not need good dexterity	Design must be operable by either left or right hand, should not require any special skills, design must indicate simple operation
15	(The device must be) operable by most ages <55	Indication of minimal effort for user such as small lever, trigger, string pull, push button
16	User must be able to operate device alone (easy)	The design must have simple operation, operable through a lever, trigger. For example if the device requires multiple operation to be performed simultaneously then the user cannot operate the device alone
17	User cannot leave wheel chair	The solution must indicate that the user does not require to get out of the chair in order to pick the fruit. For example extendable arms that allow to pick fruit,
18	Device cannot require strenuous activity	Indication of minimal effort for user such as small lever, trigger, string pull, push button
19	The device must bring joy to the user when operated	The solution must have indication of allowing user to pick the fruit from tree. For example a ramp that allows user to touch fruit.
20	Ergonomics-(The device must be) easy to use	simple operation such as trigger, lever arm, pull string, no special skills necessary to operate the device
21	(The device) must be operable by a person without technical knowledge	The design must not require any special skills to operate. Indication of simple operation such as a trigger, push button

22	Ergonomics-The use of the wheel chair should be easy to handle	The design must not interfere with the operation of wheel chair. For example if the design is too heavy such as crane it will restrict the motion of the wheel chair. If the design requires user to use both hands to operate it, he may not be able to move the wheel chair
23	The device should be ergonomical	Indication that the design is comfortable to use, no strenuous activity for user. For example the grips on the lever must be ergonomical
24	Mobility (of user) should not be lower than prior to installation of device	Indication that the design is not interfering with the normal operation of the wheel chair, could be installed on the wheel chair. Indication that the design is not too heavy making it difficult to move, should not cause tipping or falling of the user or wheel chair
25	Ergonomics -(the device must require) limited small finger movement	Indication that the device is operable using small finger movements such as pushing a button or pulling a lever, string
26	Ergonomics-(the device must require) limited wrist rotation	The mechanism to operate device is compact. Operation limited to wrist movement such as pushing a button, pulling a string
27	(The user must be able) to operate with one hand	The operating mechanism should function with one hand. Examples are press button, trigger or lever
28	Grip on device should be ergonomic	Grips should be comfortable to hold, must not have sharp edges or uncomfortable contours
29	Grip on device should not induce stress during 2 hours of use	Grip should not require strenuous movements of user, simple operation
30	Grip on device should not induce pain during 2 hours of use	Grip should not have pain causing elements such as sharp edges, uncomfortable contours or heavy effort.
31	Equipment should be quiet when in use	Must indicate absence of noise causing elements such as mating parts, electric motors
32	(The device) must not exceed a weight of 10 lbs. without fruit	Weight mentioned on the design. The design must not have any heavy parts

33	(Device's) force must be sufficient to remove fruit	The design must have mechanism that provides sufficient force to pick the fruit from the tree. For example, shears, jaws to pull the fruit etc.
34	(Device) must (allow for) use of upper body strength	Indication that the user does not need a lot of effort to operate the design. For example use of mechanisms such as lever, push buttons
35	(the device must be) strong enough	The design indicates material used is strong.
36	(the device must be) support itself	The design must be able to support itself. For example if telescopic design is used, the cross section must be large enough to support weight of the design
37	(The device must be) support up to 5 peaches at any angle	The design must have mechanism that allows picking fruit from various direction. For example a revolute joint or robotic arm
38	(The device) must be light	Indication of use of light weight material, compact design, less parts
39	Low strength required (for the device) to operate	Indication that the user does not need a lot of effort to operate the design. For example use of mechanisms such as lever, push buttons
40	Most of the weight should be near the person's body if possible	Most parts of the design are located near the user
41	Forces required to pick and gather fruit should be within acceptable range X-Y	The design has mechanism that requires less force to operate. Examples such as scissor mechanism, lever etc.
42	Force to operate (the device) should be acceptable for all size humans	Indication that the user does not need a lot of effort to operate the design. For example use of mechanisms such as lever, push buttons
43	(Device must have) range of 8-10 feet	Range of the device must be indicated on the design.
44	(The device) must store in 4 X 3 space (4 ft X 3ft, storage)	Indication such as folding design, storage size written on the design
45	(The device) must be collapsible to fit in designated space	Features such as telescopic design, retractable design, storage size indicated on design
46	(The device must be) mountable to wheel chair	Indications that the design is mounted on wheel chair. For example the design is mounted using bolts or snap fits

47	(The device) must be large enough	The solution must have storage for multiple peaches. For example cup or attached basket that is large enough to hold multiple fruits after picking
48	The design should be adjustable (8-10 ft. to 4X3 ft. storage)	Features such as telescopic design, retractable design, storage size indicated on design
49	(The device) must accommodate 2" to 4" OD spherical objects	Size indicated on the mechanism for holding or storing the fruit. Relative sizes can be used as indication if the design is drawn to scale
50	(The device) must fold to less than 4 X 3 when not in use	Features such as telescopic design, retractable design, storage size indicated on design
51	(The device must have) overall length of 5 to 7 feet	Size must be indicated on the design. Relative size can be used as an indication if the design is drawn to scale
52	Geometry-IS the device compact?(The device must be compact	The design must consist of small parts, overall design must be compact.
53	The geometry should benefit other requirements i.e. Forces, kinematics, energy	The design must have geometric features that reduce the force required to operate design. For example, lever mechanism, scissor mechanism
54	Geometry-(The device must accommodate) varying peach shapes	The design feature to hold the peaches large enough to accommodate various peach size. For example large cup or basket
55	Kinematics-(The device must accommodate) varying sizes for different arm length	The design must have some mechanism that accommodates different arm length. For example operating mechanism includes lever, scissor mechanism or pole that can extend to accommodate different arm length
56	Kinematics-(The device must accommodate) varying sizes for different peach height	The design must allow the user to pick peaches from various height ranges. For example telescopic design
57	The device must not hinder maintenance by disabled person	Indication the design will need minimal maintenance -few moving parts, simple design. The design should not require complex disassembly for maintenance purpose. Frequency of maintenance

		indicated on the design
58	(The device) must last at least 3 years	Life of the design must be mentioned on the sheet
59	(The device) must be storable on wheel chair	The design must be compact to fit on wheel chair. The design could also be mounted on wheel chair or could be folded to fit the storage on wheel chair
60	Maintenance-The device should have little maintenance	Indication the design will need minimal maintenance -few moving parts, simple design. Frequency of maintenance indicated on the design
61	Maintenance-The device should have a life time of at least 100 fruits picking	The life time of the design is mentioned on the sheet
62	(The device must have) durability	Durability of the design should be mentioned. The design is made of material that is durable
63	(The device) must be maintainable using simple hand tools	Indication that maintenance will not require the use of complex tools. For example if cleaning the design requires disassembly, then using standard fasteners in the design will require a screw driver for disassembly
64	Material-(for the device must be) reflective	Material used for the design must be mentioned and must be reflective such as metal, glass
65	Material-(for the device must be) corrosion resistant	Material used for the design must be mentioned and must be corrosion resistant such as plastic, rubber, ceramic etc
66	Material should be strong	Material used for the design must be mentioned and must be strong
67	(The device) must pick at least 2 fruits	The design must have feature that allows to hold at least two fruits at a time after picking. For example cup, mesh guard, basket
68	(The device must have) all motion caused manually	The design must not have any automation, this is indicated by absence of buttons or electrical/electronic components

69	(The device) must be usable while person is in comfortable position in the wheel chair	The operation of the design must be simple such as pushing a button or operating lever. It must not require the user to get out of the wheel chair
70	Three full-sized peaches (must) fit inside (the device)(hold fruit)	The mechanism to hold the fruit must be large enough to fit three peaches - large cup or basket, size of the mechanism may be mentioned on the solution sheet
71	(The device) must be easy to operate	simple operation such as trigger, lever arm, pull string, no special skills necessary to operate the device
72	(The device) must be easy to move to targeted peach	The design must be easy to maneuver from peach to peach on the tree. For example revolute joints that allow movement in multiple directions, telescopic design
73	(The device) must bring peach down to basket after picking	The design must have mechanism to hold the peach after it is picked and bring it to the basket. For example retractable design that folds to bring peach to basket
74	Operation-The design should have simple operation	The operation of the design must be simple such as pushing a button or operating lever. It must not require the user to get out of the wheel chair
75	(The device) must store fruits that have been picked	The design must have mechanism to store the fruit after picking. For example cup, mesh guard, basket
76	(The device) must be able to function with handicapped person	The design must be operable while sitting in wheel chair and must not require the user to get off the chair.
77	(The device must be) compatible with wheel chair	The design features must not hinder or interfere with the operation of wheel chair. For example if the design has overhangs, it may cause the wheel chair to tip off
78	(The device) must have the ability to expand from wheel chair user's hand 8 ft. from ground	The design must have mechanism that expands. For example telescopic design, scissor mechanism, folding design
79	The device should not obstruct normal wheel chair functions	The design should not interfere with other parts of wheel chair. For example if the design has overhang it may cause the wheel chair to tip over. If the design is too heavy such as a

		crane, it may prevent the user to move the wheel chair.
80	The device should not obstruct normal wheel chair operations	The design must allow user to operate the wheel chair normally. For example if the design has overhang it may cause the wheel chair to tip over. If the design is too heavy such as a crane, it may prevent the user to move the wheel chair.
81	Ergonomics-(The device) should be able to be used for more than just peaches	The mechanisms use to pick and hold the fruit should be compatible with other fruits. For example a cup of fixed size may not be used for variety of fruits but flexible jaws as a holding feature can be used for variety of fruits
82	Ergonomics-(The device should) include all fruits that grow on trees, with few exceptions	The mechanisms use to pick and hold the fruit should be compatible with other fruits. For example if cup as a holding feature can be used for variety of fruits similar in size to a peach
83	It (the device) should be able to be used comfortably for 30 minutes to an hour	The operating mechanism must not the user to input heavy effort. The mechanism may require small hand movements and no strenuous activities. For example if the lever operation requires the use to raise their hands, the user can get tired in less than 30 minutes. While if the operation only requires the user to pull a trigger, then he may be able to operate the device for longer duration
84	(The device) must be able to obtain peach in reasonable amount of time	The design must not involve complex movements to get peach from the tree. The operation must be fairly simple such as pulling levers or operating triggers to get the fruit
85	(The device) must be movable from tree to tree	The design must be compact and light weight so that it can be moved from tree to tree. Features such as wheels on the design indicate that the design is mobile. Also if the design is

		mounted on wheel chair , it can be moved with the wheel chair
86	(The device) must operate throughout harvest season	The duration for which the design can be operated must be mentioned on the sheet
87	The user can still maneuver in their chair	The design must not interfere with the user operating the chair. For example if the design requires user to use both hands, he may not be able to maneuver the chair
88	(The user) must be able to remove peach after picking	The design must have features that allow the user to remove the fruit after picking. For example, if the picked fruit is collected in a mesh guard, the user can remove the fruit to put the fruit in the basket
89	Operation -(The operation of the device must be) easy to learn	The design must have simple operating mechanism. For example if the user requires to perform more than five activities to pick the fruit, then the operation could be difficult to learn and fairly complex
90	(The device must be) built for summer conditions -high heat	The design must indicate the material it is made of. The material must be able to with stand high heat
91	(The device must be) built for summer conditions-humid	The design must indicate the material it is made of. The material must be able to with stand humid weather conditions
92	(The device) must not change overall functionality of wheel chair	The design must allow the user to operate the wheel chair normally. If the design is too heavy, it may restrict the motion of the wheel chair. Relative scale may be used to gage the size of design if it is drawn to scale.
93	(The device must) use commercially available mounting	The type of mountings used must be mentioned on the solution sheet.
94	(The device must) use commercially available actuating hardware	The type of actuating hardware must be mentioned on the solution sheet

95	(The device must) have a design suitable for ease of manufacturability	The parts of design must not have complex features that cannot be manufactured using available processes. For example if the design has complex contours, it might be difficult to manufacture.
96	(The device must) have a design suitable for ease of mass production	The design must have use of standard parts for ease of mass production. For example indication of standard size nuts, bolts, screws, gears etc.
97	Production-(The device must be produced in) mass quantity or specialized based on market size	Production quantity must be mentioned on the design
98	Fruits must pass quality control inspection for bruising	The mechanism must have fruit protecting elements such as mesh guards or cups that would prevent the damage to fruit from the point it is picked till the time it is placed in the basket
99	Fruits must pass quality control inspection for other marks	The mechanism must have fruit protecting elements such as mesh guards or cups that would prevent the damage to fruit from the point it is picked till the time it is placed in the basket
100	(The device) must not cause any damage to the peaches	The mechanism to pick and hold fruit should not damage the fruit. The mechanism must not have sharp edges or collapsible elements that can crush fruit and make it inedible. The mechanism should not let the fruit fall on ground
101	The design should not allow the fruit to fall to ground	The design must have means to hold the fruit after it is picked till the time that it is placed in the basket and thereby prevent the fruit from touching the ground. This could be a cup or funnel that puts the picked fruit in the basket. Other examples also include mesh guards, basket attached with picker
102	(The device) must not cause harm to person (Safety)	The design must not have elements that could harm the user, for example sharp edges, electrical components that could cause shock. The center of gravity (CG) of the design should be balanced so that it does not cause the

		design and/or the user to tip over
103	The design should be safe	The design must not have elements that could harm the user, for example sharp edges, electrical components that could cause shock. The design should not harm the tree or the fruit
104	Device should not damage tree	The design must not have features that could cut or damage the tree. For example sharp edges
105	(The device must) present no pinch points to the user	The design must have indication of protection against pinch points. Features such as safety guards could be used to prevent potential pinch points
106	(The device) cannot allow wheel chair to tip during operation	The design must not have over hangs or other features that can cause imbalance or tipping over while operating. The center of gravity of the design should not cause imbalance after picking the fruit
107	Safety-The device should not have potential to interfere with health issues (such as electric charge that could disrupt pacemaker)	The design must not have electrical components such as motors or control circuits.
108	(The device must have) no sharp edges etc.	The design must not have any sharp edges
109	Fruit cannot touch ground	The design must have means to hold the fruit after it is picked till the time that it is placed in the basket and thereby prevent the fruit from touching the ground. This could be a cup or funnel that puts the picked fruit in the basket. Other examples also include mesh guards, basket attached with picker
110	Device cannot damage limbs	The design must not have sharp edges that could damage the limbs

111	(The device) must not "cut" any other part of tree (for safety reasons)	The design must not have sharp parts or edges that could damage the tree. For example if the design has shears for cutting the peach then there is possibility of damage to the tree itself
112	(The device) must not "remove" any other part of tree (for safety reasons)	The design must not have sharp parts or edges that could damage the tree. For example if the design has shears for cutting the peach then there is possibility of damage to the tree itself
113	Stability of device under operation	The design must not have over hangs or other features that can cause imbalance or tipping over while operating. The CG of the design should not cause imbalance after picking the fruit
114	Stability of user under operation	The design must not have over hangs or other features that can cause imbalance of the user while operating. The CG of the design should not cause imbalance after picking the fruit
115	Failsafe mechanism	The design must have fail safe mechanism.
116	(The device must have) back-ups to prevent peach drop	The design must have guards to prevent peach drop. For example mesh or cup that would prevent peach from falling down
117	(The device must have) shields to prevent peach drop	The design must have guards to prevent peach drop. For example mesh or cup that would prevent peach from falling down
118	Transport-(The device) must not drop peaches on the ground	The mechanism to pick and hold fruit should not damage the fruit. The mechanism must not have sharp edges or collapsible elements that can crush fruit and make it inedible. The mechanism should not let the fruit fall on ground
119	(Fruit cannot be) inedible	The mechanism to pick and hold fruit should not damage the fruit. The mechanism must not have sharp edges or collapsible elements that can crush fruit and make it inedible. The mechanism should not let the fruit fall on ground

120	(Fruit cannot be)bruised	The mechanism to pick and hold fruit should not damage the fruit. The mechanism must not have sharp edges or collapsible elements that can crush fruit and make it inedible. The mechanism should not let the fruit fall on ground
121	(Fruit cannot be in) multiple pieces	The mechanism to pick and hold fruit should not damage the fruit. The mechanism must not have sharp edges or collapsible elements that can crush fruit and make it inedible. The mechanism should not let the fruit fall on ground
122	(The device must have) simple controls	simple control mechanisms such as trigger, lever arm, pull string
123	(The device) must be easily transportable	The design is not too heavy, small components, foldable design that fits in small place, can move with wheel chair