Investigating the Influence of Functional and NonFunctional Requirements on Change and Change Propagation

by

Nathaniel Conde Voris

A thesis submitted to the Office of Graduate Programs of Florida Institute of Technology in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

Melbourne, Florida December, 2017
We the undersigned committee hereby approve the attached thesis, “Investigating the Influence of Functional and NonFunctional Requirements on Change and Change Propagation” by Nathaniel Conde Voris.

Beshoy Morkos  
Assistant Professor  
Florida Institute of Technology

Chiradeep Sen  
Assistant Professor  
Florida Institute of Technology

Gnana Bhaskar Tenali  
Professor  
Florida Institute of Technology

Hamid Hefazi  
Professor  
Florida Institute of Technology
Investigating the Influence of Functional and NonFunctional Requirements on Change and Change Propagation

Author: Nathaniel Conde Voris

Advisor: Beshoy Wahib Morkos, Ph. D.

The research presented in this thesis investigates the nature of change and change propagation in requirement documents as they are influenced by requirement type, that is, “Functional” or “NonFunctional,” and how an understanding of these types may refine utilization of the Automated Requirement Change Propagation Prediction (ARCPP) tool when used by design teams.

Functionality is determined by asking the question “Does the requirement prescribe something for the project to accomplish?” This question separates the requirements into their requisite types. Using the ARCPP, the relationship and propagation scores were determined for each requirement. Furthermore, the number of relationships were determined for each requirement.

It was discovered that, generally, requirements are most closely related to others of the same type. NonFunctional requirements are the type most likely to instigate change propagation, in all cases. Furthermore, nonfunctional requirements are most likely to instigate change to whichever type of requirement is most numerous in the requirement document. Additionally, the requirements were most likely to propagate to other requirements of the same type. Finally, change propagation paths through the different
types of requirements is dependent on the number of each type of requirement in a requirement document, rather than the types of requirements, themselves.

The calculations for finding the propagation and relationship scores are included. Additionally, the method for determining the relationships between functional and nonfunctional requirements has been included to enable design teams to better predict change propagation based on both the number of relationships and the propagation scores generated.
# Table of Contents

Table of Contents ..................................................................................................... v

List of Figures ......................................................................................................... vii

List of Tables ......................................................................................................... viii

Acknowledgement ................................................................................................... ix

Dedication ................................................................................................................ xi

Chapter 1 – Introduction ........................................................................................ 1
  Motivation to Research Functional and NonFunctional Requirements .................... 5
  Intellectual Merit of the Research ........................................................................... 6
  Broader Impact of the Results ................................................................................ 7

Chapter 2 – Background on Requirements and Change Propagation ............. 11
  Relevant and Prior Work ......................................................................................... 11
  Background on Requirements ................................................................................ 12
  Current Uses of Functional Requirements ........................................................... 13
  Current Methods for Managing NonFunctional Requirements ........................... 17
  Previous Work on Change Propagation ............................................................... 21
  Automated Requirement Change Propagation Prediction Tool ............................ 24

Chapter 3 – Research Method, Approach, and Execution ................................. 34
  Research Questions .............................................................................................. 35
  Question 1 ............................................................................................................ 35
  Question 2 ............................................................................................................ 35
  Case Studies .......................................................................................................... 36
  Toho ..................................................................................................................... 39
  Pierburg ................................................................................................................. 39
  Experimental Procedure ......................................................................................... 40
  Use of Automated Requirement Change Propagation Prediction Tool .................. 45
  Data Analysis ........................................................................................................ 47

Chapter 4 – Results and Application to Research Questions ............................ 49
  Toho ..................................................................................................................... 49
  Pierburg ................................................................................................................. 56
  Summary of Results and Comparison to Actual Results ...................................... 61
  Addressing the Research Questions ...................................................................... 65
  Research Question 1 ............................................................................................ 65
  Research Question 2 ............................................................................................ 66

Chapter 5 –Recommendations, Conclusion, and Future Work ....................... 68
  Discussion of Results ............................................................................................ 73
  Recommendations for Implementation .............................................................. 77
List of Figures

Figure 1 - ECN Prediction Method [4] ................................................................. 25
Figure 2 - Example Zero (Baseline), One, and Two Path Length DSMs [4] ....... 26
Figure 3 - Graphical Representation of Example (Continued) [5] ...................... 26
Figure 4 - Requirement Change Propagation Method [4] .................................... 27
Figure 5 - Example POS Requirement Tagging [4] ............................................. 28
Figure 6 - Example Overlay DSM [5] ............................................................... 32
Figure 7 - Example ECN from Toho [4] ............................................................ 37
Figure 8 - Experimental Procedure .................................................................. 40
Figure 9 - Toho: Average Scores of Both Functional and NonFunctional
    Relationships vs Requirements ................................................................. 49
Figure 10 - Toho: Average Functional and NonFunctional Propagation Scores vs
    Requirement ............................................................................................... 52
Figure 11 - Toho: Number of Functional and NonFunctional Propagation
    Relationships vs Requirement ..................................................................... 54
Figure 12 - Pierburg: Average Scores for Functional and NonFunctional
    Relationships vs Requirements ................................................................. 57
Figure 13 - Pierburg: Average Functional and NonFunctional Propagation Scores
    vs Requirement .......................................................................................... 58
Figure 14 - Pierburg: Number of Functional and NonFunctional Propagation
    Relationships vs Requirement ..................................................................... 59
List of Tables

Table 1 - Research Questions ................................................................. 4
Table 2 - Example Tagged Requirement with Nouns, Verbs, and Keywords [4] ... 29
Table 3 - Excerpts of 215 Relators Combinations [4] .................................. 29
Table 4 - Example of a Functional and NonFunctional Requirement from Pierburg .................................................................................. 41
Table 5 - Example of Ambiguous Requirement Type from Pierburg ............. 43
Table 6 - Toho and Pierburg: Functional and NonFunctional Requirement Populations ........................................................................................................... 44
Table 7 - Toho Functional and NonFunctional Changed Requirements Breakdown ........................................................................................................... 44
Table 8 - Pierburg Functional and NonFunctional Changed Requirements Breakdown ........................................................................................................... 45
Table 9 - Toho Paired t-Test Results for Relationship Scores ......................... 50
Table 10 - Toho Paired t-Test Results for Propagation Scores ......................... 52
Table 11 - Toho Paired t-Test for Propagation Relationships ......................... 54
Table 12 - Toho Total Document Pscores .................................................... 55
Table 13 - Toho Paired t-Test for Change Level Pscores ................................ 55
Table 14 – Pierburg Paired t-Test Results for Relationship Scores .................. 57
Table 15 - Pierburg Paired t-Test for Propagation Scores ............................... 58
Table 16 - Pierburg Paired t-Test for Propagation Relationships .................... 60
Table 17 - Pierburg Total Document Pscores .................................................. 60
Table 18 - Pierburg Paired t-Test for Change Level Pscores ............................ 61
Table 19 - Comparison of Toho and Pierburg Results ..................................... 63
Table 20 - Summary of Observed Functional and NonFunctional Requirement Behaviors ........................................................................................................... 64
Table 21 – Addressed RQ1 ........................................................................ 74
Table 22 - Addressed RQ2 ........................................................................... 75
Table 23 - Proposed Research Questions for Future Work: Part 1 ................. 81
Table 24 - Proposed Research Questions for Future Work: Part 2 ................. 82
Acknowledgement

Ruthie, we’ve been friends ever since we met in 9th grade, and there’s no other friend that I have who has contributed more towards the growth of my faith and growth as a person than you. You have challenged me and forced me to think of the grey areas. You have always been there for me. Thank you for helping me to grow, for helping me to define what I believe to be true in this life. Thank you for helping me to push through this thesis. I love you, Ruthie!

Mom and Dad, you have been there to offer your advice, your support, and your encouragement from the day that I started reading until now, and I can never repay you. Thank you for leading me to Jesus and for teaching me to be the kind of man that He wants me to be and for helping me get there. If it weren’t for you both, I would be a much lesser man than I am now, and I can only hope that, when I have children of my own, I am able to give them an upbringing comparable to the one you gave me. Thank you for your patience when I didn’t want to do school. Thank you, Mom, for pushing me to excel in high school and for helping me to gear my coursework towards engineering. If it weren’t for you, I wouldn’t have been anywhere near prepared for the work that I undertook in college. Thank you, Dad, for those late nights when we would smoke a cigar, drink a beer, and you’d listen to me work through what I wanted to do with my degree. If it weren’t for you, I wouldn’t have decided to seriously pursue academia as a viable career option for both research and teaching. I love you both!

I would like to thank my advisor, Beshoy Morkos, for his support throughout this entire endeavor and for extending to me the opportunity to study under him whilst I was still an undergraduate. Without his guidance and assistance, and, perhaps most importantly, his patience, this thesis would certainly not have been written.

Additionally, I would like to thank my lab mates in STRIDE. Their levity and encouragement helped me to work through this thesis. They helped me run analyses and format my thesis, but, more than that, they have been good friends to work with on this and
other projects, and I will miss them as we separate. I will be praying for their future successes.

Finally, I would like to thank my other committee members, Dr. Chiradeep Sen and Dr. Bhaskar Tenali, for agreeing to join my Master’s committee at the last second and for all of the helpful comments and perspectives that they offered on my thesis and defense presentation. Their patience is much appreciated and I would not have refined my thesis as well without their assistance.

“This material is based upon work supported by the National Science Foundation under Grant No. 1463358. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.”
Dedication

To my wonderful parents, Brent and Rachel, and my lovely, incredible fiancée, Ruthie.
Chapter 1 – Introduction

Requirements are statements written by design teams, made up of engineers and designers, to describe the properties of a system and define the parameters by which project success can be measured. Generally, requirements are derived from customer desires or regulatory influences; however, they may be specified based on the expertise of the engineers and designers and their knowledge of the environment in which the product is intended to operate. Additionally, requirements can be derived from each other, depending on issues that arise that must be addressed. These statements are written to provide guidelines for engineers and designers to follow so that the project has a greater chance of being successful. Furthermore, they enable design teams to set milestones for themselves so that they can track their progress and report this to other stakeholders.

Requirement statements are classified in multiple ways by researchers and professionals working in industry. One way that they are classified is into the two types of “Functional” and “NonFunctional”. This classification is used to differentiate between the segments of a product that the two types are intended to address. Further discussion of functional and nonfunctional types is offered later in this thesis.

Engineering change is also a current topic of research, for example, [1]–[3]. Much of the literature regards the management of engineering change because penalties to cost and the project development timeline are incurred due to mismanaged changes during the design process. Engineering change has also been analyzed for the effects of change propagation on project success. Additionally, there has been research into the affect of requirements change on change propagation, such as in [4]–[6]. Furthermore, the research into both kinds of change and change propagation offer tools to predict changes so that they can be managed and the penalties to cost and project development timeline addressed and minimized before they occur.
This thesis presents a study on Functional and Nonfunctional Requirements during instances of change and change propagation. While the use of requirements has been previously studied [4]–[6] in change propagation, there is a research gap in understanding how the type of requirement (functional or nonfunctional) affects how change and the subsequent propagation occurs. The study presented in this thesis serves to address this gap.

Functional and nonfunctional requirements were selected as the focal points of this study because of the prevalence of this classification in the literature. Requirements are often split into this classification to differentiate between the different kinds of customer desires. Where functional requirements address, quantitatively and qualitatively, the customer needs and desires as they relate to the express purpose of a system, nonfunctional requirements address every part of a system not directly pertaining to its purpose. For example, nonfunctional requirements may address size, maintainability, recyclability, and other characteristics that a customer may desire in a system, so long as these do not directly impact what the system is supposed to accomplish. Research into how requirements influence the design process and how to address the two types such that they are less likely to cause penalties in project cost and the timeline, has been a topic that many authors in academia have addressed. Furthermore, research has been focused to one kind or the other. In particular, functional requirements have been investigated for their ability to tag information and make it easier to recall throughout a project, as well as how they are prioritized when making design decisions. Likewise, nonfunctional requirements have been explored for their influence on customer satisfaction and project success.

The kinds and strengths of relationships formed between requirements of both types, among themselves and across dimensional boundaries, and propagation scores generated using the Automated Requirement Change Propagation Prediction (ARCPP) tool were considered so that the likelihood of change and change propagation, within and across the functional/nonfunctional divide, can be found. The role of requirement functionality in change and change propagation has yet to be fully explored. Additionally, understanding change propagation will enable engineers to address the potential consequences of change
before a decision is made. For these two reasons, the research questions in Table 1 were drafted to help guide the data collection and analysis process. Research Questions 1 and 2 (RQ1 and RQ2) address the relationship between requirement type and its influence on change and change propagation. To address RQ1, an exploration for any influence on change and change propagation, whether that be significantly higher relationship and propagation scores for one type over the other or far more of one type being changed than the other. For RQ2, the actual changes that occurred in the case study projects, as well as the requirement populations of the top 15% of requirements that are most likely to change due to simulated changes to the requirements document, are investigated to find whether changes are more likely to cross dimensional boundaries or stay within them. Justification is provided for the selection of the top 15% in the research method section of the thesis.
<table>
<thead>
<tr>
<th>Research Question 1</th>
<th><strong>Question</strong></th>
<th>Does the requirement type, as classified by functional or nonfunctional, influence change and change propagation?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Hypothesis</strong></td>
<td>Requirement type, whether functional or nonfunctional, will have some noticeable influence on the change and change propagation characteristics of each requirement.</td>
</tr>
<tr>
<td></td>
<td><strong>Verification Method</strong></td>
<td>Split the two sets of project requirements into functional and nonfunctional types and analyze the relationships made among requirements as well as the population distribution of each type within the requirements documents.</td>
</tr>
<tr>
<td>Research Question 2</td>
<td><strong>Question</strong></td>
<td>Does a requirement’s status as functional or nonfunctional make it more likely to propagate change forward to others of the same type?</td>
</tr>
<tr>
<td></td>
<td><strong>Hypothesis</strong></td>
<td>There is a possibility that requirements may propagate change across functional boundaries; however, this is counterintuitive because most researchers would expect nonfunctional requirements to affect only other nonfunctional requirements, and likewise for functional requirements.</td>
</tr>
<tr>
<td></td>
<td><strong>Verification Method</strong></td>
<td>Examine the relationship and propagation scores among requirements and run statistical tests to determine the strength of relationship between the different requirement types.</td>
</tr>
</tbody>
</table>
If the hypotheses are proven, then a number of engineering applications that companies can leverage within industry, ranging from a restructuring of how requirements are written to how engineering teams can best manage and satisfy these requirements over the course of a project, may be formulated. Furthermore, the results of this study can serve as a catalyst for new research to be performed within the field of engineering requirements, engineering change, and change propagation. Should the first hypothesis be incorrect and requirement type not have a significant impact of change or change propagation, then there would not be a need to go further. However, if the first hypothesis is confirmed and the second hypothesis be incorrect and change propagation be more likely to cross dimensional boundaries, then new research may be begun into why this happens and what might be the cause for it.

**Motivation to Research Functional and NonFunctional Requirements**

Requirements have long served as the backbone to design projects. They often serve as the contract between customers and engineers. Recently, [4]–[6] have used requirements to support change propagation management. This was a major advancement in change propagation and decision science studies as it allowed engineers to make informed design change decisions before a physical architecture was realized. The next, fundamental step in understanding requirement change propagation is to determine if the type of requirement impacts this phenomenon.

To understand the nature of engineering change in design projects, it becomes necessary at some point to investigate the nature of requirements. It has been determined that requirements can be divided into two types: Functional and NonFunctional. As requirements are known to change, sometimes drastically, over the course of a design project, and because it has been demonstrated that this change can propagate through requirements lists, it would be beneficial to investigate the affect a requirement’s functionality might have on the process and result of changes.
The next step in understanding how requirements can be utilized in change propagation is to further understand how requirement type can relate to propagation. This may reveal that requirement types have an impact on propagation, which in turn may alter elicitation and management of requirements.

**Intellectual Merit of the Research**

Currently, the vast amount of research on requirement functionality has been performed with regard to software engineering. This research attempts to merge the realms of functional and nonfunctional requirements in the context of software engineering with engineering design in the physical domain. The data presented relies on past advances made in the field of requirement change propagation and some built-in functions of data analysis to generate statistically significant results that will enlighten researchers, designers, and engineers on how requirements propagate and whether these differences in type have any effect on the propagation paths and results.

The intellectual merit of this research lies in understanding how change propagation fundamentally varies between functional and nonfunctional requirements. Moreover, a cross dimensional phenomenon, of which engineers may not be aware, sought. Should requirements only propagate within their own dimension, i.e., functional to functional and nonfunctional to nonfunctional, then, to refine the propagation models that currently exist, it would be beneficial to split them into their requisite types and run the functional set of requirements through the ARCPP separately from the nonfunctional set so that relationships are more properly modeled. However, if there is some amount of cross dimensional propagation that occurs within requirements lists, then the kinds of grammatical constructions or other nuances within requirements that potentially cause the cross dimensional phenomenon might be defined and the ARCPP adapted to analyze them these cues so that only the strongest relationships between requirements are recorded and scored.
Currently, change propagation is understood in terms of requirement volatility, and research has been completed to increase the resolution with regards to this [6]. Considering both requirement functionality and the volatility metrics detailed in [6], would allow researchers to further classify requirements based on their characteristics and might, if encoded, enable computers to read requirements in their raw text format and highlight, with even greater certainty, the requirements that are most likely to change and if that change is likely to propagate to others in the list.

**Broader Impact of the Results**

The results of this research are that the relationships between functional and nonfunctional requirements, both within their own types and across the functional/nonfunctional boundary, will be understood and allow engineering teams to make more informed decisions when investigating which requirement change they ought to go with, if any at all. Understanding these relationships will allow design teams to analyze their requirements and, with additional effort on the front end, be able to see how their decisions to change one requirement or another can carry on through the requirements list, which may wind up saving them from having to do corrective work.

The style in which requirements are written might change, as was previously mentioned, depending on how their types impact change and change propagation. If there is evidence that functional requirements propagate more with other functional requirements, and likewise with nonfunctional requirements, then it may be beneficial in industry to separate the two types within the requirements document so that each kind can be examined, using the ARCPP, to find the most likely propagation paths through each list. Moreover, requirements could be written in terms of functionality at the inception of a project. If this were the case and a requirements document could be written almost entirely in functional requirements so that each one pertained to specific, emergent functions of the entire system being designed instead of some of them referring to the quality of the function, then requirement relationships might be quantified and propagation paths predicted with greater certainty. Likewise, for nonfunctional requirements: If it is discovered that more
nonfunctional requirements exist in a requirements document to begin with, perhaps it would be beneficial on the front end to rewrite functional requirements to be slightly vaguer and address quality so that the same result could be achieved.

Conversely, if it is discovered that requirement types prefer propagation across the functional/nonfunctional boundary, then propagation paths may become more convoluted because numerous switchbacks could exist where a change can vacillate from a combination of functional and nonfunctional requirements to just one or the other or a combination of the two and then back again multiple times. It would be desirable to mitigate this jumping across the boundary as close to the beginning of the project as possible, so engineers and designers may want to write requirements as specifically as they are able so that clear relationships can be drawn between the functional and nonfunctional requirements. This will reduce any ambiguity in a requirement and afford some ease to the engineers and designers when finding relationships between requirements without the use of the ARCPP, although, as in the previous case, the would be able to highlight potentially missed propagation paths between requirements and would be even more of a necessity in this instance.

The impact that requirement functionality has on change propagation may influence how they are managed within a requirements document. For example, if a nonfunctional requirement were known to have a high chance of propagating change, then it would be a simple task, if functionality impacts change propagation, to examine the propagation path and see what other nonfunctional requirements are most likely to be affected by this change. Likewise, a functional requirement that is highly likely to change could have its propagation path examined to help engineers and designers determine whether it is worth the penalties to timeline and cost that the change could incur should that decision be made. Moreover, this decision-making aid could be implemented however requirement functionality affects change and change propagation, whether that be within or across dimensional boundaries, because the propagation paths of functional and nonfunctional requirements can be determined regardless of any within or cross dimensional behavior.
Additionally, requirements might be managed by learning about whether they are functional or nonfunctional and what that means for change and change propagation is by paying particularly close attention to their relationship scores. The relationship scores that are generated throughout a requirements list may shed some further light on the propagation scores that are generated, depending on how strong the relationships formed within and across the dimensional boundary are. For instance, it may be that a functional requirement is likely to propagate to several other functional requirements; however, if the relationship scores among them all are on the smaller end of the scale, then it may be that these requirements are somehow associated with each other by a path that isn’t showing up in the relationship scores. If so, it may be wise to investigate further into what might cause that propagation and see if the requirement in question is highly related to some nonfunctional requirements that are mutual relations of the requirements that are likely to change. If this situation happens to be true, then engineers and designers can explore different options about how to mitigate the change propagation through the linking nonfunctional requirements. Additionally, this applies to nonfunctional requirements in the opposite direction. What would happen if, in the example, the relationship score among the functional requirements were high like the propagation score? In that case, the engineers and designers need not investigate other requirements to mitigate the potential change and can consider that particular propagation path.

The exploration into how requirement functionality may affect how they can be written and managed may demonstrate how satisfying them might be affected by their type. Because functional requirements have to do with the emergent function of the entire project, these often have the most weight attached to them and ought to be addressed first. However, if requirement functionality impacts change and change propagation and it is known that the functional requirement in question will cause a number of changes downstream to other, especially functional, requirements, then engineers and designers can pause and consider this potential impact to decide whether satisfying the requirement is worth the cost that will be incurred or whether it is worth talking to the customer again to gauge how important satisfying that requirement is. Likewise, nonfunctional requirements often have to do with
either the quality of the function that the project is supposed to perform or with other aspects of the project that the customer may desire that don’t necessarily affect the function; so, if a nonfunctional requirement is highly related to other nonfunctional requirements and highly likely to propagate change to them, then it may be to make that change at the beginning of the project before the other nonfunctional requirements begin to be satisfied; otherwise, time may be lost to redesigns to address any issues that come up.

A larger concern appears if nonfunctional requirements are found to be likely to propagate to functional requirements. As functional requirements are more often constraints than criteria, if a nonfunctional requirement is changed without first investigating the propagation paths and relationship scores, it may become much more difficult to satisfy a related functional requirement, and extra iterations of the design process become necessary.
Chapter 2 – Background on Requirements and Change Propagation

Requirements are an integral part of any design project, addressing customer concerns and desires and providing guidance on the kinds of solutions that can be proposed when addressing problems. Therefore, it not a surprise that so much research has been devoted to how these requirements can be defined and further categorized, how they may interact with each other, how they may interact with proposed solutions through the design process, how they can be managed, and other uses that perhaps have not been attempted.

Relevant and Prior Work

Previous work on requirements has focused on defining the specific roles that requirements have in the design process and how they are managed so that potential conflicts between them can be minimized. Besides this, research has been performed into how requirements can be classified. Currently, the two major classifications of requirements are functional and nonfunctional requirements; however, these can be further subdivided.

Functional requirements are those that have to do with the intended use of a system. These are the kind of requirement that laymen, and researchers, are most familiar with because their influence seems to be more obvious. Because they are so well known intuitively, the research regarding these requirements more often addresses novel methods of leveraging them to mitigate penalties to project cost and timelines, rather than defining what they are.

There is evidence that nonfunctional requirements, in particular, can be further categorized based on the specific project area that they may address. According to some sources, nonfunctional requirements can be separated into as many as 161 different kinds of requirements [7]. These categories have been reduced down to 27 major subgroups by [8], as will be discussed later. Furthermore, it has been demonstrated that nonfunctional
requirements are widely interconnected [9], so great care has to be taken when addressing or using them during the design process.

**Background on Requirements**

Requirements dictate the specific details of problem solutions, addressing how they are supposed to solve a problem, and how well, and what the physical dimensions of each feature are supposed to be so that end user desires, both explicit and implicit, are met. [3] states that requirements “…are a formalization of a product vision.” As such, it follows that requirements would be integral to the timeline and development cost over the entirety of the design process, from the conceptual portion of design to the actual point of manufacturing and assembly. Naturally, if the end user’s desires change, then requirements will have to change with them to ensure that they are fulfilled to as great an extent as possible.

End user desires that are expressed explicitly often take the form of functional requirements because they often address the particular task that the end user would like to accomplish. These functional requirements are often the easiest to satisfy in a project because they can be well defined. Unfortunately, end user satisfaction does not solely come from satisfying functional requirements; indeed, it has been demonstrated that projects can fail because they don’t meet the nonfunctional requirement specifications [10] that may be implicit in end user desires.

Because nonfunctional requirements are somewhat elusive and harder to define than functional requirements, the current research on them has been to propose new, and further refine existing, elicitation processes that engineers and designers use to gather them [11], manage how they are addressed in the design process [11], define their importance [12], and use them to discover potential problem areas in system designs [13]. However, nonfunctional requirements are treated differently in the literature than functional requirements. The vast majority of the research on nonfunctional requirements has been performed from a software engineering perspective; therefore, all of the research performed on nonfunctional requirements must be considered in terms of concepts instead
of concrete examples since there is not necessarily a one-to-one translation between software engineering system components and system components in the physical realm. Still, the concepts are applicable and should not be disregarded when discussing the state-of-the-art for nonfunctional requirements.

In industry applications, requirements are used to frame stakeholder desires such that they can be addressed over the course of the design process [14]. Rightly identifying them at the beginning of a project is of particular importance to a project because of their effect on its success [15]. Furthermore, since projects change over time, so, too, do their requirements [16]–[18], and because managing costs and delays associated with changing requirements is difficult [19], [20], a novel approach for predicting and managing change in requirements was needed. In [21], requirements as a whole are described as "a condition or capability to which a system must conform; either derived directly from user needs, or stated in a contract, standard, specification, or other formally imposed document."

Therefore, when writing requirements from stakeholder desires, the actual result is that definitions for the different components necessary for the realization of the system are created [22]. Furthermore, requirements both assist engineers and designers in communicating information about the project and their satisfaction helps to ensure project success [23]. Addressing them is important because, in a survey from [24], product managers reported that some of the greatest challenges faced in design were poor requirements.

**Current Uses of Functional Requirements**

Functional requirements have to do with the intended use of a system. [25] gives a working definition for functional requirements as a requirement that “defines a behavior or action that needs to be supported with the system.” Because these are constraints on what the solution is supposed to be able to perform, researchers have, in the past, focused on them more than nonfunctional requirements and the current literature has shifted to include research on new methods of utilizing them.
As functional requirements are generally well understood, there is more research into how they can be utilized in engineering design, besides just describing the intended uses of a project or system. For example, Chen et al. [26], propose a method of using functional requirements to tag and retrieve engineering knowledge throughout the design process so that it can be shared more easily among the different stakeholders.

To do this, Chen et al. [26] model functional requirements using basic descriptions that separate each into items known as functional entities (verb and noun) and modifiers (adverbs). These descriptions are then represented using semantic graph theory [27]–[29] and linked to design cases. By using the functional requirements like this, engineering knowledge over the course of designs can be called upon using the tagged functional requirements.

Further applications of functional requirements are explored in [30] which attempts system optimization by exploring the inherent conflicts within lists of functional requirements. In the design of complex systems, 70% of the cost of a product can be incurred in the early stages of product development because of design decisions; however, only 15% of the final product cost is directly attributable to design activities [31]. To decrease companies' time-to-market, more efficient design tools are needed for engineers and designers to make decisions about requirements [30].

To address functional requirements more efficiently, as in [32], they are sorted by their geometric parameters, the difference between these two papers being that [32] does not attempt to optimize like [30]. In their optimization approach, [30] consider designer preferences in a similar manner to [33]–[37] which enables them to more easily explore the potential advantages and disadvantages of deciding to address some goals over others. Furthermore, they use the structural and functional models proposed by [38] to relate functional requirements to the physical domain and help engineers and designers understand the trade-offs that may exist between the two.
Functional models, according to [39] are defined with regard to "design metrics" which are measures for stakeholder satisfaction. Whereas, functional models are created according to satisfaction, the structural models explored are based on quantitative measures, such as units of length, mass, currency, and the like. Because these two models are not built using the same terms, [40] have provided representations, known as Function-Behavior-Structure models (FBS models), to enable engineers and designers to compare the two and allow system behaviors to be modeled.

Using a measure adapted from [41], in conjunction with a weighted mean aggregation function from [42] to mitigate biases and whose weights were determined using an Analytic Hierarchy Process (AHP) from [30], [43] created a process of ranking possible solution concepts, represented by the FBS models from [40], against each other. The potential solution concepts were then optimized, based on the research on genetic algorithms developed by [44], which compared pairs of concepts and chose the one that was more satisfactory. This process can help engineers and designers in making decisions on which functional requirements are the most important to satisfy based on a design optimization centered on functional requirements in conflict with each other.

Another attempt at using functional requirements for optimization in the design process is proposed in [45]. However, the authors propose a different method from [30]. In [45], the focus is on how to make design decisions that will reduce iterations in the design process by considering how functional requirements for different solution concepts may be related to each other, along with desired performance characteristics for the concepts. There have been papers written that propose methods to decide among design concepts using performance criteria [46]–[49] and papers that explore the choices solely from a functional requirement standpoint [50], however none, yet, that consider both together.

In considering both performance specifications and functional requirements, [45] seek to address potential tradeoffs between the two by combining them into a single model that compares the relationship strength between functions and performance to assist engineers and designers in selecting solution concepts. [45] first identified six different kinds of
design information (function, concept, performance measure, physical parameter, preference, and constraint) necessary for expressing design concepts in the models. These pieces of information are then coupled and represented using an object-oriented view, like that proposed in [51], and ranked using the AHP, as proposed by [52], [53]. From here, the coupling strength of functions and constraints are both found and, along with the performance rating, which is determined by AHP [52], combined into a single mathematical function to be optimized according to the method proposed and used in [54]–[56]. Each factor is then compared pairwise through AHP [53] to define how important each is relative to each other. Engineers and designers are then able to make decisions based on how important these items are, relative to each other. [45] acknowledge that there may be many solution concepts and finding the optimized choice may be difficult, so they recommend using the Brand-and-Reduce algorithm proposed in [57], [58] to narrow down the choices.

Additionally, functional requirements are utilized as specifications and information links when defining geometric tolerances throughout the design process [59]. There has been research into how they can be used during the specification stage of the design process to begin defining tolerances earlier in the design process to mitigate potential penalties to cost and quality [32]. Generally, at the inception of the design process, there are already functional requirements that begin to specify tolerances [60].

To start managing the functional requirements that drive tolerances in the specification phase, four categories were identified [32]: design specifications, description of the user, expected performance, and others. The geometric functional requirements were split into quantitative and qualitative groups and, in [32], the qualitative functional requirements covering feature angularity, parallelism, perpendicularity, concentricity, coaxiality, symmetry, and position [61], [62] were specifically addressed. Symbols representing these qualitative functional requirements, following the standards set in [61], [62] were created so that they could be easily integrated into CAD software and engineers and designers can easily reference the geometric functional requirements for tolerances being represented in the CAD.
**Current Methods for Managing NonFunctional Requirements**

Requirements have been reliably separated into functional and nonfunctional requirements in the minds of researchers. Nonfunctional requirements are defined in the 2010 IEEE and ISO/IEC Standard 24765: Systems and software engineering - vocabulary [63] as "A software requirement that describes not what the software will do but how the software will do it." However, nonfunctional requirements are not relegated to a specific discipline, so this definition might be rewritten to serve others by dropping the word "software" and replacing it with "system". Olivier De Weck offers a definition for nonfunctional requirements, stating that they are "... properties of systems that are not necessarily part of the fundamental set of functions or constraints and sometimes not in the requirements" at all [7]. Furthermore, another definition has been offered by [64] as requirements that, as opposed to describing some task that a system must accomplish, define system properties, and are usually derived from functional requirements.

Whereas Functional requirements are written to address what objectives a system ought to accomplish, nonfunctional requirements are intended to address how that system accomplishes those tasks and the quality with which it does [8]. Because of this distinction, nonfunctional requirements are often harder to satisfy than functional requirements because the problems they define are generally not measured quantitatively. Moreover, nonfunctional requirements are made more difficult to address for two other reasons: they have been known to relate to each other strongly, and their importance is relative to each system [65].

In [8], Adams mentions that there has been work completed to compile lists of the different kinds of nonfunctional requirements, undertaken by [7], [65], [66]. Altogether, [7], [65], [66] noted 161, 38, and 19 kinds of nonfunctional requirements, respectively. To address the variety of categories that these kinds of nonfunctional requirements define, it was necessary to establish some additional models to attempt to reduce the total number of nonfunctional requirement categories from 208 to 27.
To reduce them, Adams employed seven separate models and an international set of standards created between 1976 and 2011. These models are Boehm's Software Quality Initiative [67], Cavano and McCall's Model [68], McCall's and Matsumoto's Factor Model Tree [69], Software Quality Evaluation Guidebook [70], FURPS and FURPS+ [71], Blundell, Hines and Stach's Quality Measures [72], Somerville's Classification Schema [73], and the international standard ISO/IEC Std 25010 [74]. Using the models prescribed above, the total list of nonfunctional requirement types was reduced from 208 to 27 that were further divided into four areas, altogether called the NFR Taxonomy: System Design Concerns, System Adaptation Concerns, System Viability Concerns, and System Sustainment Concerns [8]. From these areas, nonfunctional requirements can potentially be addressed more easily because those that are grouped together are necessarily more closely related to each other.

Alternatively, nonfunctional requirements have been managed in the literature by using Bayesian Belief Networks (BN), the arguments for which can be found in [75], to model scenarios in which the nonfunctional requirements are satisfied by Functional requirements. In this approach, nonfunctional requirements are seen as subjective, quality-based goals to be attained by satisfying the more concrete functional requirements to which they are related [76]–[78]. Furthermore, they are assumed to incorporate the effects of the human side of requirement fulfillment [13].

Previously, using scenarios to model requirement relationships and estimate potentially problematic relationship paths had been proposed by [79]–[81] and the idea further defined by [82], [83] and explored by [11], [84]–[86]. This perspective enables engineers and designers to take into account human elements when investigating how to satisfy nonfunctional requirements by incorporating languages like i* [78] and TROPOS [87]; however, it should be noted that the latter does not work with nonfunctional requirements directly. In [88], this approach was further refined by the implementation of a software program intended to ease its use. This program was able to run a number of scenarios based on an initial one supplied to it by engineers [88]; however, it was discovered that too
many scenarios were generated when it ran and the engineers and designers who were still required to review them all were overwhelmed with data.

Further developments in this program are what led to the incorporation of BNs. In particular, BNs are used to account for both technology and people when attempting to find solutions to satisfy nonfunctional requirements. The program takes inputs in the form of scenarios and, using BNs, assesses the probabilities of different failure modes throughout the test scenarios, reporting both those that pass based on a predetermined standard, and how well the system passed in the scenarios and highlighting potential causes of failure. Furthermore, it can represent the results of the analyses graphically and in a tabular form, similar to ALBERT II [89], KAOS [90] and GRAIL [83], or TROLL [91] so that it is more accessible to the customer. Because it is costly to set up the BNs, it is intended for systems where multiple design iterations are expected so that the cost of several tests can help to justify them [13].

When errors are caused by nonfunctional requirement mismanagement, they can incur cost and be effort-intensive to fix, and, according to [92]–[94], are the most expensive to address; furthermore, they may extend the design phase and cause the system being designed to be delayed in reaching deployment phase. According to [95], eliciting Nonfunctional requirements is often futile because they get ignored, stakeholders may not be able to define them well enough so they can be addressed, the standard process of requesting them from stakeholders does not help, and there is a lack of agreement on nonfunctional requirement importance in terms of what they mean and how they are useful for a system. [65], [82], [96] have recently attempted to demonstrate that they can be addressed earlier in the process to mitigate any risk of delay and cost penalties.

In [11], a process is proposed, based on the Language Extended Lexicon (LEL) [97], that enables engineers and designers to represent nonfunctional requirements in models created using Unified Modeling Language (UML). This is different from other research into nonfunctional requirements which focus on the final product and how nonfunctional requirements should work for it [98]–[102]. Other authors, however, [65], [98], [103],
[104], have presented processes for addressing nonfunctional requirements. This approach, using LEL, actually helps to prepare nonfunctional requirements for use in scenario modeling, like in [13], and attempts to assist engineers and designers in building NFR graphs [76], which reduce them into concepts from the more nebulous ideas that they can begin as.

The problems in implementing the proposed method in [11] arise because building the nonfunctional requirement graphs from the LEL was not automated. This allows differences between the graphs, as engineers and designers must necessarily build them subjectively since humans cannot be perfectly objective. The time spent on creating the nonfunctional requirement graphs, too, can be cause for apprehension with some engineers and designers and discourage them from using the approach because more complex projects would require the creation of more graphs. Furthermore, [11] states that the process is leveraged on nonfunctional requirements that express tasks that the product must perform, that is, those that are closer to functional requirements; although, it notes that the process may help in addressing nonfunctional requirements that express product standards for safety, traceability, performance, and accuracy.

Research has additionally been performed on addressing both functional requirements and nonfunctional requirements at the same time. For instance, in [12], Mohammad Dabbagh and Sai Peck Lee propose a method by which functional requirements and nonfunctional requirements might both be prioritized at the same time in software system design. Furthermore, two other approaches exist to address this issue: Analytic Hierarchy Process (AHP) [105], where requirements are put into a square matrix subjectively scored, in pairs, from 1 to 9 to determine priority, and Hybrid Assessment Method (HAM) [106]. Additionally, AHP has been adapted by other researchers into such methods as Case-Based Ranking (CBRank) [107] and Cost-Value Approach [108]. Besides these adaptations, other authors have designed their own processes for requirement prioritization. Wiegers [109] proposes a prioritization based on four concerns, benefit, penalty, cost, and risk, each scored from 1 to 9. [110] allows stakeholders a larger role in prioritization by connecting requirements to business values in Value-Oriented Prioritization (VOP). [111], [112] each
propose a partially automated prioritization tool implemented at the requirement elicitation phase and a correlation-based priority assessment where requirements are investigated for relationships, respectively. Finally, the authors of [113] propose an applied genetic algorithm to prioritize requirements, especially for larger projects.

In [12] the idea for prioritization of the Functional requirements and nonfunctional requirements is called Integrated Prioritization Approach (IPA). Building on the research performed by [9] which discovered that, in a software system, nonfunctional requirements can influence all functional requirements in that same software system, the IPA [12] builds a decision matrix that relates the two types of requirements to each other by asking the question of how important each nonfunctional requirement is to each FR and requesting a score of the importance from the engineers and designers. The approach then takes these scores and converts them into scaled values that are subsequently analyzed using triangular fuzzy numbers. The final rankings for the nonfunctional requirements are determined based on the results of the analysis while the functional requirements are ranked based on weighted averages that come from that analysis. When compared to AHP and HAM, IPA was a quicker tool to use and yielded similar results [12].

**Previous Work on Change Propagation**

Change propagation has been an area of interest for researchers for some time now because of the penalties in cost and project timeline that can occur when it goes unaddressed. There has been research starting from engineering change itself, e.g. [1], to using tools to address how to prioritize changes from an engineering change and requirements perspective [2], [4]–[6], and others.

In [1], Jarratt et al. investigate engineering change starting from the perspective that engineering changes can be defined as “changes to parts, drawings, or software that have already been released during the product design process…, regardless of the scale of the change.” The authors of this paper approach change, instead of as something to be avoided, as something that needs to be managed as a part of the innovation process. In [114], companies were surveyed as to their perspective on engineering change. Although 82% of
them acknowledged that it was necessary for innovation, and innovative products needed to for greater income, few of them were able to manage changes well, not knowing either the product components that might be affected or how the overall product might be affected.

In addressing engineering change so that it can be managed and its impacts more well known by engineers and designers, Jarratt et al. [1] describe, first, engineering change itself, second, the process by which engineering changes are performed and, third, some tools in the literature that exist to help manage these changes.

[1] noted that engineering change can generally be characterized as either “emergent,” that is, coming from something in the project to address an error, safety [115], change of function, or quality problems, or “initiated,” which refers to those changes that originate from some external source like customers [116], the sales and marketing department [115], [116], product support, production, suppliers [117], product engineering, company management, or new laws being passed by legislators [118]. Additionally, other categories based on the urgency [119], [120] and timing [121] of the change have been proposed. These are changes that are either immediate, mandatory, or convenient for the urgent categories or early, mid-production, or late/expedited for the timing categories.

Some of the impetus for making decisions on engineering changes quickly comes from the Rule of 10 from [122], [123] that estimates an average increase by a factor of 10 of the implementation cost as each design process phase passes. Obviously, then, it is important to address changes early in the design process, especially because, as time passes, more stakeholders become involved and information must be shared among more people. This can lead to a lack of information for some parties and additional strife to the cost of implementation [116], [117].

A lack of addressing the previously mentioned issues is a potential cause for change propagation; however, there are three other items that may cause it. Reported by [124], there are three kinds of high-level relationships that may cause change to propagate. These
are couplings between: components and manufacturing, components within a subsystem, components in different subsystems.

Managing engineering change, then, is a necessary activity for engineers and designers to undertake. Jarratt et al. [1] present some further material on how to accomplish this and divide the support tools into two kinds: those that help manage workflow and those that help decision-making. For those that help the decision-making process, one that is used in particular is the Change Prediction Method (CPM). The CPM uses two Design Structure Matrices (DSMs) to model the risk of change by taking into account the likelihood and impact of that change [125].

Additional research using DSMs to manage change propagation has been performed by Giffin et al. in [2]. This work in particular informs the work on change propagation by Morkos [4], Hein [5], Menon [6], and others. In [2], a greater understanding of change propagation is sought from the Engineering Change Notices (ECNs) that a company generates over the course of a project. The change propagation explored therein is a higher-level analysis of how ECNs are related to each other across component and system boundaries. Furthermore, DSMs are utilized here, and analyzed using graph theory tools, to model the paths by which change propagation might occur. Additionally, other observations from this paper include those that change magnitude and occurrence are inversely related, that is, larger changes appear far fewer times in ECNs than do minor changes, and change propagation can be divided into two kinds, “Initiated” (intended) and “emergent” (unintended) [2]. These changes are propagated, as determined by Morkos [4], Hein [5], and Menon [6], through agents known as that are carried through by such agents characterized as “robust,” “transmittance,” “multiply,” and “absorb”.

The research performed by both Hein [5] and Menon [6], provide a deeper understanding of how requirements impact change and change propagation. Where previous work done by Morkos [4] investigated the ability of the ARCPP in predicting change propagation, Hein [5] refined the tool by exploring the individual abilities of the physical (nouns) and functional (verbs) domains in predicting change propagation. What was discovered in his
work is that the physical domain predicts change propagation more reliably than the functional domain does; however, the functional domain adds resolution to the potential propagations predicted by the relators in the physical domain.

Menon’s work in [6] uses a different method for the prediction of change and change propagation from Morkos and Hein. In his work, Menon successfully predicts change propagation in requirements using complex network metrics and validates his results against those already proven in the ARCPP work by Morkos and Hein. His conclusion, however, is that the prediction capabilities of the complex network metrics method varies based on the network models used to represent requirements since requirement documents are unique and the analysis must be tailored to each.

**Automated Requirement Change Propagation Prediction Tool**

The ARCPP tool, previously developed by Morkos [4] and refined by Hein [5], was validated by successfully predicting the propagation of change due to initial changes made to requirements through the design process within two projects undertaken by the same company. The process for predicting the change propagation is depicted in Figure 1. This process makes use of engineering change notifications (ECNs) to signal when changes have occurred. When a stakeholder discovers a needed change, the change process is begun and a formal ECN is drafted that details the exact items to be changed, when the change was requested, when it will take place, and any penalties to the project timeline and cost so that both the company and customer might approve it [4].
To validate the ARCPP from the ECN forms, they were examined to determine which requirements were to be involved in the change. Additionally, the requirements were organized within a relationship path length Design Structure Matrix (DSM), as depicted in the workflow in Figure 1, previously validated by Morkos to explain requirement relationships consistently, to determine the relationships that existed among the requirements [4]. This kind of DSM depicts the shortest path lengths available in requirement relationships. Furthermore, multiple DSMs can be constructed to illustrate different degrees of relationship between requirements. Figure 2 is an example of how requirements are illustrated to be related to each other through a DSM. If a change is made to Requirement E, propagation may occur between that initially changed requirement and Requirement C, which in the second figure is colored red and depicted as one path length away from Requirement E. Further propagation may then occur between Requirement C and Requirement D, which is pictured in yellow in the final part of the figure. From these two steps, it is determined that while Requirement E and Requirement D are not directly related,
change may still propagate between them because of their path length two relationship through Requirement C. A graphical illustration of these relationships is represented in Figure 3 from [5].

The physical components and functions, that is, nouns, verbs, and statistically selected keywords from each requirement, are used in [4] to construct the DSM. The company provided all of the documentation that described the ECNs and the propagations associated
with them. In validating the ARCPP, the changes confirmed to have propagated via the company’s documentation were used. Figure 4 details the process by which the ARCPP predicts change propagation through requirements: (1) requirement documents are parsed into their requisite parts of speech, using the Stanford Part of Speech Tagger, to identify the relevant tags for the analysis, (2) potential relationships between requirements are determined from the previously identified relators, (3) optimally combining the relators and determining the actual relationship propagations, and (4) ranking the likelihood of propagation to requirements.

The relating nouns and verbs were tagged in each requirement sentence using the Stanford Part of Speech tagger. This software tool takes a requirement document input in natural language and determines which part of speech each word in a requirement sentence is. It does this by utilizing an algorithm based on the Penn Treebank set that alleges to have an accuracy of 97.86% when tagging [126]. [127] gives a compilation of the parts of speech,
and their respective abbreviations, that the Stanford Part of Speech tagger can use. Below is an example from [5] of a requirement before and after being analyzed by the POS tagger.

**Requirement:** The yarn combs shall have polished stainless steel round pins mounted on zinc plated steel bar.

**POS output:** The/dt yarn/nn combs/nns shall/md have/vb polished/vbn stainless/jj steel/nn round/nn pins/nns mounted/vbd on/in zinc/nn plated/vbn steel/nn bar/nn ./.

Singular and mass nouns are tagged “NN” while their plural forms are tagged “NNS”. Verbs have a number of different tags from “VB” for the base form, to “VBD” for a past tense verb. Additionally, “VBG,” “VBN,” “VBP,” and “VBZ” refer to the gerund, past participle, non-3rd person singular, and 3rd person singular forms of verbs. Figure 5 depicts a graphical model of how a sentence is tagged.

![Figure 5 - Example POS Requirement Tagging](image)

During the part of speech tagging process, the first five nouns and verbs, in the order of their appearance in the requirement sentence, were identified as relators. Furthermore, five keywords from each requirement sentence were selected by mechanical engineering students.
as being representative of each, and these selections were statistically confirmed for accuracy [4]. Table 12, from [4], gives a requirement with its tagged relators as an example.

Table 2 - Example Tagged Requirement with Nouns, Verbs, and Keywords [4]

<table>
<thead>
<tr>
<th>Requirement</th>
<th>All switches must have labels on both the switch/connector and a permanent label on the closest mechanical fixture to the switch’s mounting. These labels should be metal tags and the permanent tag must be riveted in place.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nouns</td>
<td>Switch, label, connector, metal, tag</td>
</tr>
<tr>
<td>Verbs</td>
<td>Have, mounting, riveted</td>
</tr>
<tr>
<td>Keywords</td>
<td>Switch, label, tag, connector, fixture</td>
</tr>
</tbody>
</table>

Requirement relationships are built from optimal combinations of the relators identified in the tagging process. There are, however, \(2^n\) combinations that can be created from relators, where \(n\) is the combined number of nouns, verbs, and keywords. Each of the DSMs that was created using the combinations was then examined to determine its effectiveness in relating the requirements. Table 3 reports some of the possible combinations of the fifteen relators potentially involved in forming each DSM.

Table 3 - Excerpts of \(2^{15}\) Relators Combinations [4]

<table>
<thead>
<tr>
<th>Combination No.</th>
<th>Noun 1</th>
<th>Noun 2</th>
<th>Noun 3</th>
<th>Noun 4</th>
<th>Noun 5</th>
<th>Verb 1</th>
<th>Verb 2</th>
<th>Verb 3</th>
<th>Verb 4</th>
<th>Verb 5</th>
<th>Keyword 1</th>
<th>Keyword 2</th>
<th>Keyword 3</th>
<th>Keyword 4</th>
<th>Keyword 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1952</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6142</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17995</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27657</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Three different filters were applied to the DSMs to determine their effectiveness in relating requirements. These are the Path Length Filter (1), the Difference Between Average Path Lengths for True Positives and False Positives (2), and the Ratio of Average Path Lengths for True Positive and False Positive Relationships (3) [4].

The Path Length Filter (F1) is applied to determine that the propagated and initiating requirements being examined are actually related and whether they lie within the specified path length [4]. As an example of how this works, if a path length of 3 is specified, only the DSMs of the combinations of relators that form relationships between both the initiated and propagated will be considered [5].

Equation 1 shows how Filter 2 (F2) is calculated [4].

\[
F_2 = \frac{1}{TP} - \frac{1}{FP}
\]

**Equation 1 - Filter 2 (F2) [4]**

Here:  
\( TP \) = Average Path Length of the True Positive Relationship  
\( FP \) = Average Path Length for False Positive Relationship  
\( F_2 \) = Score for Filter 2

When using F2 to filter the DSMs, a positive score is desired. An example from [5] is given where the average path length for true positive relationships, TP, is 1.67, and the average path length for false positive relationships, FP, is 2.2. In this case, \( F_2 = \frac{1}{1.67} - \frac{1}{2.2} = 0.59 - 0.45 = 0.14 \). Because the F2 score for this example is positive, the associated combination is determined to be acceptable because the true positive relationship average path length is shorter than the false positive relationship average path length. To further refine the search for suitable DSMs, according to this example, only those with F2 scores that are greater than 0.14 should be chosen. Furthermore, it is noted that the reciprocals of the average path lengths are used in this calculation to keep F2 between the values of zero and one [5].
Equation 2 shows how Filter 3 ($F_3$) is calculated [4].

$$F_3 = \frac{\frac{1}{TP}}{\frac{1}{FP}}$$

**Equation 2 - Filter ($F_3$) [4]**

Here:

- $TP$ = Average Path Length of the True Positive Relationship
- $FP$ = Average Path Length of the False Positive Relationship
- $F_3$ = Score for Filter 3

Whereas combination effectiveness is defined in $F_2$ as a positive score, $F_3$ scores describe something different. An $F_3$ score greater than one denotes that there is a greater separation in average path lengths between true positive relationships and false positive relationships.

Hein gives another good example for how $F_3$ is used to further refine choosing between suitable combinations identified by $F_2$ [5]. In this example, two DSMs are analyzed. In the first DSM, $TP = 2$ and $FP = 2.5$. Therefore, according to Equation 1 and Equation 2, $F_2 = 0.10$ and $F_3 = 1.25$. The second DSM in this example has $F_2 = 0.10$ (just as it did previously) and $F_3 = 1.14$ (both from $TP = 1.25$ and $FP = 1.42$). What is observed is that $F_2$ for both the first and second DSMs is equal (0.10); however, because $F_3$ is greater for the first case ($1.25 > 1.14$), the first DSM is more suitable for use than the second and further refinements in combinations should seek to have an $F_2$ score of 0.10 or greater and an $F_3$ score of 1.25 or greater. Filtering combinations by these three methods leaves only the most effective DSMs available for predicting change propagation [5].

The ranking for each requirement relationship in the filtered DSMs was determined from its relationship path length [4]. Path lengths were given scores of 9 for a path length of 1, 6 for a path length of 2, and 3 for a path length of 3. Additionally, the filtered DSMs were further combined. An example of this combination is illustrated in Figure 6. These scores were then further analyzed via a root-mean-squared (RMS) approach following the formula reported in Equation 3 [4].
As Equation 3 shows, RMS was chosen because the weightings of requirement relationships is preserved in the scoring mechanism, with the impact from stronger relationships between requirements being preserved throughout the analysis[4]. Furthermore, an RMS approach was chosen over any other power because a purely summative mean does not preserve the impact of relationships, and using exponents greater than two diminishes the sensitivity of the mean to strength of relationship [128].

$$S_{\text{propagation}} = \sqrt{\frac{W_{R1}^2 + W_{R2}^2 + W_{R3}^2 + \cdots + W_{Rn}^2}{n}}$$

Equation 3 - Propagation Score as Calculated via RMS Formula [4]

As explained in [5], here:
- $S$ = Propagation score for potentially propagated requirement
- $W_{Ri}$ = weighting of previous requirement with $i$ based on relationship path length
- $n$ = number of previous requirements used in the analysis

This propagation analysis ranks requirements based on their likelihood to change, therefore, a term is needed to report how efficiently the tool is working. Review depth indicates how
much of a requirement document had to be searched to find the propagated requirement, and, as such, is the metric used to determine the efficiency of the ARCPP analysis [4]. [5] gives an example for its use in that, if the 10th highest requirement scored out of 100 requirements propagates, then the review depth is equal to 10%. This means that whomever is examining the requirements for potential changes need only investigate among the top ten requirements to find it. Thus, the efficiency of the ARCPP is determined to be greater the lower it is [5]. A more detailed description of the exact, prior analyses is reported in [4].
Chapter 3 – Research Method, Approach, and Execution

In determining the effect of a requirement’s type, whether functional or nonfunctional, on the change propagation characteristics of a requirements list, the relationships among the requirements were studied according to Research Question 1 (RQ1) and Research Question 2 (RQ2), introduced in Table 1 in Chapter 1. Requirements lists from two different projects, Toho and Pierburg, were prepared and analyzed using tools described in this thesis.

Addressing RQ1 and RQ2 requires that the author understand past research on managing requirements, change, and change propagation. As was explored in Chapter 2, requirements are divided into functional and nonfunctional types that describe both the express purpose of any designed object (functional) and how the object will perform it (nonfunctional) [63]. Furthermore, since functional requirements are defined in the public mind, more research has been performed on nonfunctional requirements, especially from a software engineering perspective.

Furthermore, change and change propagation have been explored by [3], [2], [1], [6], [5], and [4]. In particular, Morkos’ work on change and change propagation [4] serves as the starting point of the study presented here. Indeed, the requirements documents taken as case studies for this research had been previously analyzed by both [4] and [5] to validate the ARCPP tool as a means of accurately predicting requirement relationships and potential change propagations through requirement documents. Additionally, [6] utilized the requirements documents to begin characterizing individual requirements based on their complexity metrics.
Research Questions

Starting from the previous work developing the ARCPP by [4] and [5], the two research questions listed in Table 1 in Chapter 1 were posed to define the scope of the research. Because functional and nonfunctional requirements are already known to be related to each other, and have been studied by [9] and [12] from a software perspective and their interactions managed, and the relationships between requirements are what enable change to propagate through requirements lists [5], [4], the next step is to define the effect that requirement functionality has on change and change propagation, and it was to that end that RQ1 and RQ2 were posed.

Question 1

RQ1: Does the requirement type, as classified by functional or nonfunctional, influence change and change propagation?

In addressing RQ1, the Toho and Pierburg requirements are separated into the functional and nonfunctional types. This was a subjective task in which the author examined each requirement to determine its status as either functional or nonfunctional. By doing this, the population breakdowns of each requirements list were created so that the relationships between and propagation paths between requirements can be examined.

Furthermore, the results for the two projects are already known. Since the functional and nonfunctional requirements are all known, then the population of the results can be examined for whether one type greatly outnumbers the other. This will lead to further insight into the nature of change propagation and whether requirement type influences it.

Question 2

RQ2: Does a requirement’s status as functional or nonfunctional make it more likely to propagate change forward to others of the same type?

Addressing RQ2 requires that the Toho and Pierburg requirements have their relationships and potential propagations scored and the top potential changes examined for the
functional and nonfunctional populations. To generate the propagation scores for each requirement, changes are simulated to each one and the propagation scores calculated from the relationship scores generated by the ARCPP [5], [4] using the root-mean-squared formula prescribed in Equation 3. The propagation scores are then sorted based and the top 15% of potential propagations for each requirement changed are noted. These top potential changes are what makes up the review depth mentioned in [5], [4].

By examining the functional and nonfunctional requirement population breakdowns of the top potential changes and comparing them to the requirement instigating the change, insight into the likelihood of change propagation, within or across dimensional boundaries, can be gained.

Case Studies

The two projects included engineering documentation from the design process for each. Each one reported engineering change notifications (ECNs), with dates for proposal, implementation, and/or rejection, and more [4]. In [4], Morkos examined the requirements lists for both the Toho and Pierburg projects against the ECNs and manually determined change propagation. He then verified that the ARCPP was able to predict these changes using the requirements lists and their respective relationships [4]. Additionally, subsequent work by [6] and [5] was able to utilize the data from the requirements lists. They were found to be suitable for use in the case studies for the research presented here, as well. Because both projects are so different in terms of budget, requirements lists, and end products, it is the author’s goal that the results of these case studies are generalizable to requirements documents at large.

The changes in each of the projects have already been noted in [4] as being initiated by ECNs, an example of which is given in Figure 7.
Figure 7 - Example ECN from Toho [4]

According to Morkos [4], changed requirements were associated with ECNs after the fact since they are not explicitly listed on each. Morkos examined each ECN for the kind of
change being proposed and then associated it with the requirements list to determine which requirements were affected. Moreover, the requirements estimated to have changed were confirmed with the company via contact with personnel and reviewing of documentation. Change propagations, especially unexpected ones, were found to have occurred when there was evidence that an ECN had been completed and then readdressed afterwards. In Morkos’ research [4], and, subsequently, in that of Hein [5] and Menon [6], only propagated changes were explored. As such, any changes considered by the author are either those confirmed in [4]–[6] or the ones simulated to generate the propagation scores.

Two case studies were used in this thesis to increase the robustness of the results. It is simple to draw conclusions for a single set of results; however if two sets of results agree, then the conclusions are reasonably well supported. Furthermore, having two different data sets allows for more accurate results to be found. The requirement makeup, reported in Table 6, Table 7, and Table 8, of the Toho and Pierburg projects helps to illustrate how different projects, even under the same overarching company, can differ. If the two projects indicate different results given the same analyses, it can be assumed that more research has to be pursued in that area. However, if they both agree, then, because of the great differences in makeup of requirement type, the results may be generalized through to other projects and recommendations for implementation may be made to those in industry.

In performing the case studies, the first step was to prepare each requirements document by examining each requirement individually and determining whether it was functional or nonfunctional. Each of the requirements lists was then analyzed for occurrences of each type and percentages of functional and nonfunctional requirements in each list were reported for the main body of requirements and the industry-verified changes. The next step was to run each of the requirement documents through the ARCPP, developed by Morkos in [4] and further refined by Hein [96] and Menon [6], to get the relationship scores. The propagation scores, as has already been mentioned, were generated after simulating changes to the requirements document and using the relationship scores within the root-mean-squared formula from [4]. Finally, both sets of scores were separated by requirement type and the top scoring requirements examined. Furthermore, both the
relationship and propagation scores were analyzed for significant differences in scores recorded and number of relationships among requirements.

**Toho**
The Toho project, from which 159 requirements were provided, was to create spools of yarn on creels meant to hold multiple spools [4]. The project itself cost the company over two million dollars and required contributions from fifteen different personnel, ranging from businesspersons to engineers [4]. Because these 159 requirements had been previously analyzed in work completed by [4], [5], and [6] and were found to be suitable for validating the ARCPP, the differences between physical and functional domains in requirements, and complexity metrics for them all, they were deemed suitable for validating the research posited in this paper because it builds specifically off of their work.

In this research, the Toho requirements were manually split into the functional and nonfunctional types and their relationship and propagation scores found using the ARCPP and analyzed. The results are reported in Chapter 4 – Results and Application to Research Questions.

**Pierburg**
Additionally, the Pierburg project, from which 214 requirements were generated and recorded, was undertaken by the company to produce assembly lines intended to manufacture multiple versions of exhaust gas recirculation bypass flaps [4]. The total budget for this project was 1 million dollars [4]. Furthermore, these 214 requirements had been analyzed by [4], [5], and [6] and found to validate the research performed by each. Because this project, too, was used in three different research topics where each was able to be validated, the associated requirements were, again, deemed to be suitable for the research presented here.

Like the requirements from Toho, the Pierburg requirements were split into functional and nonfunctional types and then analyzed using the same process. The results of the Pierburg requirements are reported in Chapter 4 – Results and Application to Research Questions.
Experimental Procedure

Figure 8 below illustrates the process used to perform the experiment on the previously mentioned case studies. Each of the sections is described more fully in the following text.

To conduct the research, the requirements from both the Toho and Pierburg projects were first prepared and then entered into the ARCPP tool that had been previously developed by Morkos, Hein, and Menon and are documented in [4], [5], [6]. To generate the data for analysis the requirements had to be prepared and an experiment, separated into two parts, was required. In the first part of the experiment, the Toho and Pierburg requirements were entered into the ARCPP with the actual changes so that the relationship scores among all of
the requirements could be generated. In the second part, a script was written to ingest the relationship scores, simulate changes to requirements, and output the propagation scores of related requirements.

Because [4], [5], [6] have already been written using the requirements data from Toho and Pierburg, each of the requirement sets had already been parsed by the Stanford Part of Speech Tagger. This made the task of further preparing them simpler because the requirements’ different parts of speech had already been denoted. Having the requirements separated allowed for more obvious visibility of the active verb in each sentence so that the intent of the requirement could be determined.

Separating the requirements into their types required the author to examine each requirement individually to determine its affect on each project. Table 4 gives an example of two requirements reported in [4]–[6] as having changed in the project. In particular, Requirement 25 is a requirement to which change was propagated due to an initial change to Requirement 138.

Table 4 - Example of a Functional and NonFunctional Requirement from Pierburg

| Req. 25 | supplier will design equipment for fast change over time (5 minutes or less total line change over time) using quick change out tooling and fixtures. | NonFunctional |
| Req. 138 | transport pallet shall be used to transport the product pallet will not be used as fixtures for critical operations pierburg inc must approval deviations from this specification if pallet are used as fixtures then each measurement must have a capability (cpk) > 1.033 the measurement report has to be provided to pierburg inc. | Functional |

When determining the functionality of requirements, the author asked one primary question: Does the requirement prescribe something for the project to accomplish? Answering positively to this question resulted in the determination that a requirement was functional, whereas answering negatively would indicate the requirement is nonfunctional.
Therefore, when examining Requirement 25 above, because nothing is prescribed for the project to accomplish, it is determined to be nonfunctional. Additionally, Requirement 138 was examined with regard to whether it prescribed a task to the project. Because the transport pallets are being used to transport items as a part of the manufacturing/assembly process, it is determined to be functional. Each requirement in the Toho and Pierburg projects were examined individually to determine functionality.

In cases where it was ambiguous which type a requirement was, the author would leave the requirement until the end and re-examine it then, determining the functionality using the first instinct that came. For example, when examining Pierburg Requirement 176 reproduced in the Table 5, the author experienced some doubt about its functionality. To address this, the author determined that the requirement could be regarded as functional based on the first instinct upon re-reading the requirement. The author’s rationale for selecting functional as the type was because, in this case, the author considered the operator intervention to be a part of the function of the assembly line for the products. Other individuals may regard the operator as outside the system and decide, because of that, to assign the nonfunctional type to this requirement. The other difficult situations encountered like this were handled similarly.
Table 5 - Example of Ambiguous Requirement Type from Pierburg

<table>
<thead>
<tr>
<th>Req. 176</th>
<th>Functional</th>
</tr>
</thead>
<tbody>
<tr>
<td>if no part tracking (i.e. pallet rf tag, etc.) is used or if the last station (packaging) includes a process (e.g. screw driving) then process failures will require an operator intervention (e.g. a reset, password, or key switch) before the failed part can be removed from the station the operator or equipment will remove non-conforming product at the point of failure to a lock box or equivalent quality device the equipment must confirm the placement of the non-confirming material in the lock box before restarting furthermore the control has to count each failure in order to validate the number of bad parts in the box and the number of occurred failures the lock box will have ergonomic access such as a chute or gravity conveyor no part damage is allowed and has to be considered for the design of the conveyor (opportunity for potential rework) for end of line testing stations only automated bad part handling is allowed.</td>
<td></td>
</tr>
</tbody>
</table>

Because the procedure to determine requirement functionality relied on the author’s knowledge of the project and interpretation of whether the requirement would affect the designed function, the result of the preparation was subjective. To determine the type of each requirement, it had to be examined individually, with the focus primarily centered on the verb involved and, secondarily, the grammar of the sentence, to determine whether it directly influenced the function of the end result of each project or whether its affect was more nebulous. Eventually, each of the requirements list were split into the two types of “Functional” and “NonFunctional”. Table 6 shows the breakdown of the two requirement types in each of the projects. It is noticed that, even though both projects were undertaken by the same company because of their nature, they had different amounts of nonfunctional requirements compared to functional requirements and this may help to inform the results of the experiment.
Table 6 - Toho and Pierburg: Functional and NonFunctional Requirement Populations

<table>
<thead>
<tr>
<th></th>
<th>Toho</th>
<th>Pierburg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>159 Total Requirements</td>
<td>214 Total Requirements</td>
</tr>
<tr>
<td>NonFunctional</td>
<td>91 NonFunctional Requirements</td>
<td>83 NonFunctional Requirements</td>
</tr>
<tr>
<td></td>
<td>57%</td>
<td>39%</td>
</tr>
<tr>
<td>Functional</td>
<td>68 Functional Requirements</td>
<td>131 Functional Requirements</td>
</tr>
<tr>
<td></td>
<td>43%</td>
<td>61%</td>
</tr>
</tbody>
</table>

Table 7 and Table 8 report the breakdown of requirements involved in the changes, as documented by the company and reported in [4]. It can be seen that, in each case, there are different amounts of functional and nonfunctional requirements involved on each side of change, initial and propagated. In these tables, the requirement numbers of each change are recorded with their functional or nonfunctional type listed in the same position below them. The Initial Change column indicates those requirements involved in the initial change and whose change then propagated to those listed in the Propagated Change column.

Table 7 - Toho Functional and NonFunctional Changed Requirements Breakdown

<table>
<thead>
<tr>
<th></th>
<th>1st Propagation Instance</th>
<th>2nd Propagation Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement Number</td>
<td>91, 93, 96</td>
<td>[6, 110, 111]</td>
</tr>
<tr>
<td>Functionality</td>
<td>Func, Func, Func</td>
<td>NonFunc, Func, Func</td>
</tr>
<tr>
<td>Requirement Number</td>
<td>[6, 91, 93, 96, 110, 111]</td>
<td>112, 113</td>
</tr>
<tr>
<td>Functionality</td>
<td>NonFunc, Func, Func, Func, Func</td>
<td>NonFunc, Func</td>
</tr>
<tr>
<td>Requirement Number</td>
<td>Pierburg Initial Change</td>
<td>Propagated Change</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td></td>
<td>[2, 17, 22, 76, 138, 176]</td>
<td>[17, 25]</td>
</tr>
<tr>
<td>Functionality</td>
<td>[NonFunc, NonFunc, Func, NonFunc, Func, Func]</td>
<td>[NonFunc, NonFunc]</td>
</tr>
<tr>
<td>1st Propagation Instance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirement Number</td>
<td>[2, 17, 22, 25, 76, 138, 176]</td>
<td>[85]</td>
</tr>
<tr>
<td>Functionality</td>
<td>[NonFunc, NonFunc, Func, NonFunc, Func, Func]</td>
<td>[Func]</td>
</tr>
<tr>
<td>2nd Propagation Instance</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Use of Automated Requirement Change Propagation Prediction Tool

After having prepared the requirements by showing their status as either functional or nonfunctional, they were entered into the ARCPP, along with their documented changes, and the first part of the experiment was begun so that their relationship scores could be determined. These scores were generated based on how many steps apart each requirement was away from the one to which it was related. A score of 0 indicates either a lack of relationship or a relationship with more than three path lengths between the two requirements; a score of 3 means that the two requirements are related with a path length of 3; a score of 6 means that the two requirements are related with a path length of 2 between them, and a score of 9 means that the two requirements being examined are located one path length away from each other [4]. Each requirement is scored in relation to every other requirement and the three most suitable DSMs are overlaid into an n x n matrix that is created to report all of the scores, which, because of the overlay, range from 0 to 27. This matrix was then output to be further manipulated by finding the average relationship scores and standard deviations for functional requirements, nonfunctional requirements, and both together.
Once all of the relationship scores from both Toho and Pierburg were generated, the functional/nonfunctional statuses were correlated and graphs of the relationship strengths among all of the requirements were created.

To calculate the propagation score of each requirement, and start the second part of the experiment, a script was written to simulate a change to each requirement and, according to the previously discussed equation given by Morkos in [4], generate the propagation scores of each requirement.

Actual implementation of this formula assumes an initial change to a requirement. The weights of all requirements previously listed are then, according to the formula, squared, summed, and subsequently divided by their number. Finally, the square root of the number is taken to arrive at the propagation score. The script read in the relationship score matrix and calculated the propagation score for each requirement in a column and then would move on to the next column to do the same thing. Because of the RMS approach, the highest number score that a requirement can receive is 27, while the lowest is 0. A score of 27 would indicate that there is a strong likelihood that that requirement might change due to the simulated initial change. A score of 0 would indicate that the change would not propagate based on an initial change.

Again, these scores were put into an n x n matrix (in the case of the Toho project, 159 x 159, and, in the case of the Pierburg project, 214 x 214) and analyzed in this form.

These scores were sorted first by the requirement that was simulated to have initially changed and then sorted based on the propagation scores of the top 15% of influenced requirements. For Toho, this mean that the top 24 scoring requirements for each change were pulled out and put into a separate spreadsheet, and, for Pierburg, 32 requirements. The average propagation scores and standard deviations for every requirement were generated and total, weighted averages and standard deviations for the entirety of each scored sheet were found. These scores were then normalized and used in the t-Tests that were performed on the data.
Data Analysis

In analyzing the data, a paired t-test for means approach was used. Both paired and unpaired t-tests exist; however, there is a difference between them in the analyses that they perform. Paired t-tests are used when analyzing multiple tests using the same subject. In the case of this research, each subject is a separate project, either Toho or Pierburg.

The paired t-test approach applied in this research is analogous to using a single player to test a number of baseball bats, swinging both right-handed and left-handed. Using a single player ensures that the swinging motion of the player while using each bat is more consistent between trials because it is the same test subject. Using an unpaired approach would mean that the same number of players are involved as the number of bats and each player is given one bat. The results of the tests of all the players using their respective bats are then compared in the unpaired t-test versus the results of the single player using all the bats in the paired t-test. This analogy is not perfect because there are other factors, such as player fatigue and environmental conditions that affect performance, but it accurately represents how the Toho project is a single subject with each requirement representing a separate trial instead of each requirement being a subject. The result, however, is that the P-values generated when analyzing requirements within a subject indicate, generally, statistical significance, but this is still a more accurate analysis of the data than an unpaired t-test would allow.

Additionally, after the relationship and propagation analyses on the scores generated from the ARCPP, the amount of each type of requirement appearing in each project was examined from a probabilistic perspective at both the project level and at each instance of change. Furthermore, this probability score was also determined for each type of requirement, that is, functional and nonfunctional. Equation 4 for binomial probability was used to determine the probability scores.
\[ P_{score} = \sum_{x=1}^{n} \frac{N!}{x! (N-x)!} \pi^x (1-\pi)^{N-x} \]

**Equation 4 - Binomial Probability Formula for Determining the Probability Score of Changes**

Where, in this formula:

- \( N \) = Total number of requirements in a project
- \( x \) = Number of successes; from 1 to the amount of requirements of the specified type per change/project
- \( \pi \) = Probability of requirement type in question being chosen
- \( n \) = Total amount of requirements of the specified type per change/project

Therefore, in applying Equation 4, the total number of trials varied between Toho and Pierburg, as did the number of potential successes and the probability of choosing either type of requirement. For example, if applying the Pscore to the totality of the Toho project, examining specifically the functional requirements, \( N = 159, x = 68, \) and \( \pi = \frac{68}{159} = 0.42767296 \). In this case, then, the Pscore was generated for \( x = 1 \) to \( x = 68 \). Furthermore, the Pscore is cumulative; therefore, each Pscore value from \( x = 1 \) to \( x = 68 \) was summed to get the final probability of any requirement being functional in the Toho project as approximately 53.3%.

This approach of examining the Pscores of functional and nonfunctional requirements in projects was applied both to Toho and Pierburg at the project level and to each simulated change in each requirement document. Furthermore, because each simulated change is related to a different amount of each requirement type, the probability of change propagation to either type can be estimated through the use of only the number of each type of requirement related to the change and the formula given in Equation 4.
Chapter 4 – Results and Application to Research Questions

After generating the relationship scores and performing change propagation analysis on the data sets for both Toho and Pierburg requirements, they were analyzed based on their requirement types to address the aforementioned Research Questions. This section will detail the analysis performed for each Research Question with respect to the projects. The cumulative results of the analysis are used to address the research questions.

Toho

In Toho, after the requirements were split into the functional and nonfunctional types, the relationship scores are generated, averaged, and ranked according to how high each requirement scored. The chart in Figure 9 shows all of the average scores for both the functional and nonfunctional requirements.

![Toho: Average Scores For Func/NonFunc Relationships vs Requirement](image)

Figure 9 - Toho: Average Scores of Both Functional and NonFunctional Relationships vs Requirements
From Figure 9, it is noticed that nonfunctional requirements, on average, had higher relationship scores than did functional requirements. The higher relationship score shows that nonfunctional requirements form stronger relationships with other requirements than functional requirements do. However, it is noted that there are areas on the chart that indicate a higher average functional score. These requirements, according to Figure 9, form stronger relationships with functional requirements than nonfunctional requirements. The evidence that nonfunctional requirements form stronger relationships than functional requirements should cause engineers and designers to examine closely those relationships as possible paths for change propagation, given an initial change.

In addition to examining the relationship scores as they appear on a graph, the scores were analyzed using the Paired Two Sample for Means t-Test analysis. The scores were investigated from three different perspectives: All Requirements, Functional Requirements, and NonFunctional Requirements; Table 9 shows the averages of these analyzed scores.

<table>
<thead>
<tr>
<th>Table 9 - Toho Paired t-Test Results for Relationship Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="table.png" alt="Table" /></td>
</tr>
</tbody>
</table>

In the category “All Requirements,” the scores under column “Functional” report the average relationship score of every requirement to its relationship with functional requirements. Like the scores under “functional,” the scores under “NonFunctional” depict the average relationship score from every requirement to only the nonfunctional requirements. For “Functional Requirements,” the average scores reported are from only
functional requirements to every other functional requirement (under “Functional”) and to every other nonfunctional requirement (under “NonFunctional”). The same format is used for the “NonFunctional” category.

Considering all of the average scores, as was already reported in Figure 9, it is observed that the nonfunctional requirements score higher on average than other requirements. The strongest relationships, according to this table, are formed between nonfunctional requirements and other nonfunctional requirements, while the weakest relationships are formed between nonfunctional requirements and functional requirements.

Upon further inspection of Table 9, it is evident that nonfunctional requirements form the strongest receiving relationships overall, that is, the second requirement in a relationship chain has a higher relationship score if it is nonfunctional rather than functional regardless of the first requirement in that chain. However, it is additionally noticed that functional requirements form stronger relationships with each other. Therefore, the strongest relationships are within the boundaries of functionality as opposed to cross-dimensional. Furthermore, the P-value for the scores tabulated in Table 9 is less than 0.05. Therefore, this indicates that the difference in relationship scores for functional and nonfunctional requirements is reliable. This means that Toho indicates the strongest relationships exist within functional boundaries rather than cross-dimensionally.

The observation that there is a difference in relationship scores between functional and nonfunctional requirements initially indicates confirmation of the hypothesis that requirement functionality impacts change and change propagation. However, before that conclusion can be fully confirmed, more evidence about propagation itself must be presented. To that end, the propagation scores for functional and nonfunctional requirements in Toho were averaged, and normalized. Figure 10 illustrates the normalized average propagation scores for both functional and nonfunctional requirements as stacked areas, and Table 10 reports the values of the Paired Two Sample for Means t-Test that was performed thereon.
In the results reported in Table 10, it is observed that nonfunctional requirements are more likely to change, especially due to a change to a prior, related nonfunctional requirement. Additionally, functional requirements are more likely to propagate change to nonfunctional requirements instead of to other functional requirements. Because the difference in scores
is statistically significant (P-value less than 0.05), it is concluded that, based on the Toho results, nonfunctional requirements are the most likely to change due to propagation. Furthermore, the higher score for a functional requirement propagating to a nonfunctional requirement suggests that functional requirements are less constrained than nonfunctional requirements when propagating change.

Comparing the ranking results between the relationship and propagation scores, disparities appear in how functional requirements react to a propagated change. In particular, functional requirements are more closely related to each other than to nonfunctional requirements; however, propagation is more likely to occur to a nonfunctional requirement than a functional requirement, regardless of the functionality of the initiating change.

Further insight into the nature of change propagation may be gleaned from examining how many requirements of each type to which functional and nonfunctional requirements propagate. Figure 11 shows the number of propagation relationships that each requirement has. As observed from this graph, more propagations for all requirements regardless of functionality are to nonfunctional requirements rather than to functional requirements. Likewise, the tabular summary of this data, Table 11, shows that the majority of propagation relationships are to nonfunctional requirements, regardless of the initiating requirement.
Figure 11 - Toho: Number of Functional and NonFunctional Propagation Relationships vs Requirement

Table 11 - Toho Paired t-Test for Propagation Relationships

<table>
<thead>
<tr>
<th></th>
<th>All Requirements</th>
<th>Functional Requirements</th>
<th>NonFunctional Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Func</td>
<td># NonFunc</td>
<td># Func</td>
</tr>
<tr>
<td>Mean</td>
<td>5.648 ± 3.059</td>
<td>18.35 ± 3.059</td>
<td>5.485 ± 2.888</td>
</tr>
<tr>
<td>P-value</td>
<td>2.347E-59</td>
<td>3.686E-28</td>
<td>1.228E-32</td>
</tr>
</tbody>
</table>

Because the results of the propagation relationships are statistically significant, the conclusion that nonfunctional requirements are the most likely to receive change propagation can be drawn.
Finally, the probability of change propagation to either type of requirement was studied in Toho, the results of which are reported in Table 12 and Table 13.

### Table 12 - Toho Total Document Pscores

<table>
<thead>
<tr>
<th>Total Requirements Document</th>
<th>Pscore Functional</th>
<th>Pscore NonFunctional</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5335</td>
<td>0.5304</td>
</tr>
</tbody>
</table>

### Table 13 - Toho Paired t-Test for Change Level Pscores

<table>
<thead>
<tr>
<th></th>
<th>All Requirements</th>
<th>Functional Requirements</th>
<th>NonFunctional Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Func</td>
<td>NonFunc</td>
<td>Func</td>
</tr>
<tr>
<td>Mean</td>
<td>0.5840 ± 0.0509</td>
<td>0.5873 ± 0.0154</td>
<td>0.6015 ± 0.0117</td>
</tr>
<tr>
<td>P-value</td>
<td>0.5239</td>
<td>8.361E-18</td>
<td>0.0356</td>
</tr>
</tbody>
</table>

In this analysis, based on the amount of each type of requirement in a given change, overall (probabilities given in Table 13), and the amount of each type in the whole requirement document (probabilities given in Table 12), both functional and nonfunctional requirements will change with similar probability; however, this result is not statistically significant because the P-value is less than the threshold of 0.05. Further examining the requirements from the perspective of functional boundary, functional requirements have a higher probability of propagating change to other functional requirements by nearly 1.5% and nonfunctional requirements have a higher probability of propagating change to other nonfunctional requirements by nearly 1.9%. Furthermore, each of these results is
statistically significant (P-value = 8.361E-18 < 0.05 and P-value = 0.0356 < 0.05, respectively).

From a statistical perspective, then, despite the fewer numbers of functional requirements overall in the Toho project, and especially within individual changes, the probability that a change may propagate from one functional requirement to at least one other functional requirement is slightly higher than the probability that the change may propagate from that same functional requirement to at least one nonfunctional requirement. Additionally, the probability of change propagation from a nonfunctional requirement to at least one other requirement is slightly higher if the type of requirement receiving the propagation is also nonfunctional than if it’s functional.

Pierburg

The same analyses were performed on the Pierburg data. Figure 12 shows the average relationship scores from functional and nonfunctional requirements. For the Pierburg project, more requirements were of the functional type than were nonfunctional.
Figure 12 - Pierburg: Average Scores for Functional and NonFunctional Relationships vs Requirements

It is observed that the overall average scores for relationships between functional requirements and all others are slightly greater than those for nonfunctional requirements. This indicates that, here, functional requirements form stronger relationships overall than do nonfunctional requirements. The total breakdown of requirement average relationship scores is reported in Table 14.

<table>
<thead>
<tr>
<th></th>
<th>All Requirements</th>
<th>Functional Requirements</th>
<th>NonFunctional Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Func</td>
<td>NonFunc</td>
<td>Func</td>
</tr>
<tr>
<td>P-value</td>
<td>2.750E-12</td>
<td>3.008E-27</td>
<td>4.727E-03</td>
</tr>
</tbody>
</table>

As in Table 9 for Toho, the relationship scores for Pierburg are statistically significant with P-values less than 0.05. Furthermore, the strongest requirement relationships are within the functional boundaries. However, like was mentioned with regard to Figure 12, the functional requirements of Pierburg score higher overall than the nonfunctional requirements.

The Pierburg results indicate that requirements are most closely related to others of the same type; however, functional requirements are more strongly related, overall.

Considering the propagation scores, Figure 13 gives the average propagation scores for the Pierburg requirements in stacked area form, and Table 15 reports the normalized average propagation scores for functional and nonfunctional requirements in Pierburg.
According to Table 15, the difference in propagation scores, though small, is statistically significant and indicates that the functional requirement type is the most likely, overall, to change due to propagation. Furthermore, functional requirements are most likely to change due to propagation from a nonfunctional source.
Figure 14 shows the number of functional and nonfunctional propagations per requirement, and Table 16 gives the average numbers of propagated requirements per type.

![Figure 14 - Pierburg: Number of Propagations for Func/NonFunc Changes vs Requirement](image)

**Figure 14 - Pierburg: Number of Functional and NonFunctional Propagation Relationships for vs Requirement**

It is observed from both Figure 14 and the statistically significant results reported in Table 16 that, unlike the Toho project, more functional requirements are connected through potential propagation paths than nonfunctional requirements.
Table 16 - Pierburg Paired t-Test for Propagation Relationships

<table>
<thead>
<tr>
<th></th>
<th>All Requirements</th>
<th>Functional Requirements</th>
<th>NonFunctional Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Func</td>
<td># NonFunc</td>
<td># Func</td>
</tr>
<tr>
<td><strong>P-value</strong></td>
<td>3.711E-129</td>
<td>1.773E-90</td>
<td>1.860E-43</td>
</tr>
</tbody>
</table>

Finally, the probability of change is explored through the quantity of each requirement type involved in the whole of the Pierburg requirement document and at the individual change level. The results of this analysis are given in Table 17 and Table 18.

Table 17 - Pierburg Total Document Pscores

<table>
<thead>
<tr>
<th>Total Requirements Document</th>
<th>Pscore Functional</th>
<th>Pscore NonFunctional</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5259</td>
<td>0.5300</td>
</tr>
</tbody>
</table>
Table 18 - Pierburg Paired t-Test for Change Level Pscores

<table>
<thead>
<tr>
<th></th>
<th>All Requirements</th>
<th>Functional Requirements</th>
<th>NonFunctional Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pscore Func</td>
<td>Pscore NonFunc</td>
<td>Pscore Func</td>
</tr>
<tr>
<td>Mean</td>
<td>0.5669 ± 0.0013</td>
<td>0.5868 ± 0.0069</td>
<td>0.5667 ± 0.0012</td>
</tr>
<tr>
<td>P-value</td>
<td>5.887E-118</td>
<td>1.140E-82</td>
<td>1.131E-39</td>
</tr>
</tbody>
</table>

In examining the results given, it is observed that, when approaching change from the standpoint of the whole document, at least one nonfunctional requirement is approximately 0.4% more likely to change than at least one functional requirement because of quantity despite functional requirements being more numerous in this project. Furthermore, examining the results of individual changes from a global perspective, the statistically significant result reported in Table 18 shows that change propagation to at least one nonfunctional requirement is 2% more probable than change propagation to at least one functional requirement. Furthermore, this is broken down so that, from a functional perspective, change propagation to at least one nonfunctional requirement is 2% more likely than change propagation to at least one functional requirement from an instigating functional requirement. Additionally, the same phenomenon occurs with an instigating nonfunctional requirement, and these results are statistically significant (P-value = 1.140E-82 < 0.05 and P-value = 1.131E-39 < 0.05, respectively).

**Summary of Results and Comparison to Actual Results**

Recalling the actual requirements involved in the change instances in Toho and Pierburg, listed in Table 7 and Table 8 respectively, it is observed that, in Toho, despite the greater number of nonfunctional requirements than functional requirements, functional requirements were more involved in the actual changes. Furthermore, in Pierburg, where
functional requirements outnumbered nonfunctional requirements, the requirement types were evenly split between the two.

According to the relationship scores from Toho and Pierburg, reported in Table 9 and Table 14 respectively, functional requirements and nonfunctional requirements are more closely related to other requirements within their functional boundaries than they do across functional boundaries. Furthermore, the propagation scores from Toho and Pierburg, reported in Table 10 and Table 15 respectively, suggest that, in Toho, nonfunctional requirements are the most likely to propagate change to both types of requirements and that nonfunctional requirements are most likely to receive change from both nonfunctional and functional instigating requirements. Additionally, in Pierburg, nonfunctional requirements instigate change propagation most strongly to functional requirements, and functional requirements also instigate change propagation most strongly to other functional requirements. The number of relationships at the individual change level were also examined for each project and the results given in Table 11 for Toho and Table 16 for Pierburg. For Toho, these results report that, at each instance of change, the instigating requirement is related to far more nonfunctional requirements than functional requirements, regardless of its own functionality. Conversely, for Pierburg, the results report that, at each instance of change, the instigating requirement is related to far more functional requirements than nonfunctional requirements, regardless of its own functionality. Finally, the examinations of probability of change propagation to either type of requirement from the perspective of how many of each type are included in a requirement document are reported in Table 12 and Table 13 for Toho and Table 17 and Table 18 for Pierburg. The global results for the Toho requirements report that there is no great bias for change propagation to either type of requirement, and, further, investigating the probabilities at the individual change level, within functional boundaries there is a slight bias towards requirements of the same type. In the case of Pierburg, the global probability indicates that there is a slight bias towards nonfunctional requirements; additionally, there is a slight bias at the individual change level towards propagation to nonfunctional requirements.
Comparing the Toho and Pierburg results yields Table 19.

**Table 19 - Comparison of Toho and Pierburg Results**

<table>
<thead>
<tr>
<th></th>
<th>Toho</th>
<th>Pierburg</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Relationship Scores</strong></td>
<td>Requirements are related to each other most strongly within their functional boundaries.</td>
<td>Requirements are related to each other most strongly within their functional boundaries.</td>
</tr>
<tr>
<td><strong>Propagation Scores</strong></td>
<td>Propagation scores indicate that nonfunctional requirements are the most likely to propagate change, and that they are the most likely to receive propagation from either type.</td>
<td>Propagation scores indicate that functional requirements are most likely to receive propagation from either type, and that nonfunctional requirements are most likely to instigate propagation.</td>
</tr>
<tr>
<td><strong>Propagation Relationships</strong></td>
<td>NonFunctional requirements form the most potential propagation relationships, regardless of instigating requirement type.</td>
<td>Functional requirements form the most potential propagation relationships, regardless of instigating requirement type.</td>
</tr>
<tr>
<td><strong>Pscore</strong></td>
<td>Globally, there is no major bias of propagation toward either requirement type; however, at the individual change level, there is a slight bias of propagation towards functional requirements.</td>
<td>Globally, there is a slight bias of propagation towards nonfunctional requirements, and there is a slight bias of propagation, at the individual change level, towards nonfunctional requirements.</td>
</tr>
</tbody>
</table>

Finally, a summary of observed behaviors for functional and nonfunctional requirements, derived from Table 19, is given in Table 20.
Table 20 - Summary of Observed Functional and NonFunctional Requirement Behaviors

<table>
<thead>
<tr>
<th>Relationship Scores</th>
<th>Functional Requirements</th>
<th>NonFunctional Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Functional requirements are most strongly related to other functional requirements.</td>
<td>NonFunctional requirements are most strongly related to other nonfunctional requirements.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Propagation Scores</th>
<th>Functional requirements are most likely to propagate change to the most numerous requirement type in a document.</th>
<th>NonFunctional requirements are most likely to propagate change to the most numerous requirement type in a document.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Propagation Relationships</th>
<th>Functional requirements form the most potential propagation relationships with the most numerous requirement type in a document.</th>
<th>NonFunctional requirements form the most potential propagation relationships with the most numerous requirement type in a document.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Pscore</th>
<th>Functional requirements have a slight bias of probability of propagation towards the less numerous requirement type in a document.</th>
<th>NonFunctional requirements have a slight bias of probability of propagation towards the less numerous requirement type in a document.</th>
</tr>
</thead>
</table>

When comparing the observed behaviors listed in Table 20 to the actual change instances in Toho and Pierburg, it is noticed that the behaviors estimated by the experimentally determined results do not match directly with what actually happened. In Toho, changes occurred primarily among functional requirements, with only two nonfunctional requirements involved at all. Furthermore, both these nonfunctional requirements first appeared because of change propagation to them. In Pierburg, five nonfunctional and four functional requirements are involved in both instances of propagation.

According to the evidence from the experiments on Toho and Pierburg, both functional and nonfunctional requirements, despite the strength of relationship to other requirements being strongest within functional boundaries, should drive change and change propagation to whichever type appears more in a requirement document. However, when examining the actual changes, change propagation in Toho primarily involves functional requirements, which appear in fewer numbers than nonfunctional requirements, and Pierburg involved slightly more nonfunctional requirements, despite nonfunctional requirements being
outnumbered by functional requirements. Because the Pscore for both functional and nonfunctional requirements indicates that the probability of change propagation is slightly biased, in all cases, towards the less numerous requirement type in a requirement document, perhaps the probability of change, based on the amount of each type of requirement in a requirement document, in combination with the propagation score, more profoundly influences how change is propagated through requirements.

Addressing the Research Questions

Based on the results above, conclusions about RQ1 and RQ2 can be drawn that increase engineers’ and designers’ understanding about change and change propagation. These conclusions can be leveraged in industry by providing insight into the importance of requirement type as it relates to change and change propagation.

Research Question 1

RQ1: Does the requirement type, as classified by functional or nonfunctional, influence change and change propagation?

In addressing this question, the relationship scores and propagation scores for both the functional and nonfunctional requirements are examined. These scores are split up by type to determine which type generally formed stronger relationships, which type had more relationships, and which was more likely to change due to propagation. In determining these items, discoveries may be made about the effect that requirement functionality has on change and change propagation.

In examining the relationship scores reported in Table 9 and Table 14, the following phenomenon is noticed. The relationships formed between requirements are strongest when applied within the same functional boundary. Furthermore, the cross-boundary relationship scores are weakest. These scores are statistically significant based on the P-values for each case being less than 0.05.
Additionally, the propagation scores were inspected for trends and it is observed that, in Toho, nonfunctional requirements score highest and, in Pierburg, functional requirements score highest. Furthermore, the highest scores are propagated from a nonfunctional requirement to others. For Toho, this means that the highest propagation score is from a nonfunctional requirement that is initially changed to a nonfunctional requirement to which change propagates. Likewise, for Pierburg, the highest propagation score is from a nonfunctional requirement that is initially changed to a functional requirement to which change propagates.

Based on the evidence presented, addressing RQ1 suggests that requirement functionality impacts change and change propagation. Although requirement relationships are strongest within functional boundaries, nonfunctional requirements ignore these boundaries when propagating change and affect both functional and nonfunctional requirements.

**Research Question 2**

**RQ2: Does a requirement’s status as functional or nonfunctional make it more likely to propagate change forward to others of the same type?**

RQ2 is addressed by further examining the propagation scores of each project and investigating the functionality of the simulated and actually propagated requirements. As when addressing RQ1, the scores were separated into each type, averaged, and ranked. Furthermore, 15% of the top requirements likely to receive change propagation were investigated. For Toho, this means that the 24 highest scoring requirements were examined, and, for Pierburg, the top 32 requirements were examined. Each of these sets of requirements were ranked individually based on their propagation scores and were averaged.

It was discovered from Table 11 and Table 16 that the Toho requirements propagated mostly to nonfunctional requirements and that the Pierburg requirements mostly to functional requirements, and, because the P-value is less than 0.05, this result is statistically significant. This means that, when examining requirement documents,
requirement type should be considered when predicting change propagation. If a document contains predominantly functional requirements, then it follows that change may propagate mostly to requirements of this type. Conversely, if a requirement document were populated with more nonfunctional requirement, it is expected that propagation will be to more nonfunctional requirements.

Furthermore, these propagated requirements align with the propagation scores reported in Table 10 and Table 15, which are also statistically significant. Similarly to the propagation relationships, the scores, in conjunction with the knowledge that functional requirements make up a greater percentage of the requirement population in Pierburg, indicate that change propagation may be a function of population size rather than something in the requirements. Likewise, the nonfunctional requirement population of Toho is greater than the functional requirement population, and this, combined with the statistically significant results of the propagation scores, indicates that the functionality of the greater portion of a requirement document is the main factor that contributes to propagation.

Table 7 and Table 8, contrary to the evidence presented, report that more functional requirements than nonfunctional requirements are involved in change propagation in Toho, and there are almost even numbers of functional and nonfunctional requirements involved in change propagation in Pierburg. The probability of propagation indicates a slight bias toward the less numerous type of requirement in a requirement document. This bias combined with the propagation scores, may explain why more functional requirements were involved in change and change propagation in Toho and why slightly more nonfunctional requirements were involved in change and change propagation in Pierburg.
Chapter 5 – Recommendations, Conclusion, and Future Work

Requirements documents are one of the most important elements of the design process. These statements translate stakeholder desires into both quantitative and qualitative guides that engineers and designers use in decision making so that a project is more likely to satisfy the project stakeholders. As such, they have been researched for their impact on project success.

Furthermore, requirements have been categorized in the literature. Two of the most popular categories are the types of “Functional” and “NonFunctional.” Each of these types addresses a certain kind of stakeholder need. Typically, functional requirements are those that expressly address the purpose of a product. They prescribe tasks to the product and define product properties directly related to accomplishing those tasks. Furthermore, they have been defined in [25] as a requirement that “defines a behavior or action that needs to be supported with the system.” Functional requirements are often received directly from the stakeholders, but may also originate from the design team as challenges are met and addressed. Conversely, nonfunctional requirements generally address external influences on a product and how well a product accomplishes the tasks prescribed by the functional requirements. Because of this, nonfunctional requirements are often derived from the external sources of influence, such as regulatory committees, or from functional requirements. Additionally, there is some flexibility in the definition of nonfunctional requirements with such statements as “A software requirement that describes not what the software will do but how the software will do it,” from [63], “… properties of systems that are not necessarily part of the fundamental set of functions or constraints and sometimes not in the requirements,” from [7], and requirements that define system properties instead of describing tasks for the system, from [64]. In the literature, research has been performed on the additional uses that functional requirements might have in the design process, and on other methods for managing nonfunctional requirements.
In particular, functional requirements have been explored for use in tagging chunks of engineering knowledge in a project so that engineers and designers might be able to search and find relevant information quickly when communicating with each other and with stakeholders [26]. Additionally, functional requirements have been used to address system optimization problems where multiple functional requirements may be in conflict with each other [30]. In performing this optimization task, functional requirements are sorted based on geometric parameters and related to the physical domain so that engineers and designers can understand the tradeoffs among the requirements when deciding which is the best choice for the success of the project [30]. These functional requirements are then related by functional models, created according the design metrics for stakeholder satisfaction [39], that are combined with structural models in the Function-Behavior-Structure model [40] that relates function to structure. Further applications for functional requirements exist to ease the design decision-making process by coupling them with performance specifications [45] and analyzing them with the AHP [52], [53] and Branch-and-Reduce algorithms [57], [58] to make the final, optimized decision. Finally, functional requirements have been used to define geometric tolerances and linked to them [59] using symbols, specified in [61], [62], so that the tolerances can be addressed using CAD software and engineers and designers can reference the associated functional requirements.

Management methods for nonfunctional requirements have been investigated for implementation in the design process, as well, because, just like with functional requirements, mismanagement of nonfunctional requirements can incur penalties to cost and be difficult to fix; furthermore, mismanagement of nonfunctional requirements is determined by some authors to be the most expensive issue to address [92]–[94]. However, to address the issue of management, nonfunctional requirements were first separated into 208 different categories of requirements by [7], [65], [66] and reduced to 27 by [8]. This method was intended to decrease the amount of effort required to sort nonfunctional requirements and address them in the design process. NonFunctional requirements have also been managed by [75] using BNs. Using BNs requires viewing nonfunctional requirements as subjective, quality-based goals to be attained, not by satisfying other
nonfunctional requirements, but by satisfying the functional requirements to which they are related [76]–[78]. Using this method also enables engineers and designers to use scenarios to model their systems to determine optimal solutions; however, this part of the analysis is only recommended for those systems where multiple iterations are expected, as BNs are costly to set up [13]. Another approach [11] to management of nonfunctional requirements is the use of the LEL [97] by representing them using UML. This approach is advantageous because it can be used in conjunction with the scenario modeling scheme [13] mentioned previously and can reduce them into NFR graphs [76] where concepts may be concretized. Finally, research has been performed into possible methods for prioritizing and managing nonfunctional requirements simultaneously with functional requirements [12]. The approach (IPA) espoused in [12] builds from research first performed by [9]. Decision matrices are created that relate the two types of requirements to each other. Using subjectively determined importance scores from engineers and designers that are subsequently converted to scaled values by the IPA, the paired functional and nonfunctional requirements are analyzed using triangular fuzzy numbers to determine the final rankings and recommend to engineers and designers which requirements are the most important to address first.

Change and change propagation have also been studied by researchers who have proposed methods of addressing it to minimize the costs, both to budget and timeline, they incur. Engineering change has been investigated by Jarratt et al. [1] as an entity that can be addressed. They define it as “changes to parts, drawings, or software that have already been released during the product design process…, regardless of the scale of the change.” Primarily, Jarratt et al. view engineering change as a necessary part of the innovation process that needs managing so project success is more likely to be achieved, and not something to avoided. Furthermore, engineering change can be characterized as “emergent” or “initiated.” Emergent engineering changes arise to address an error, safety [115], change of function, or quality problems. Initiated engineering changes are begun from some source external to the project like customers [116], the sales and marketing department [115], [116], product support, production, suppliers [117], product engineering,
company management, or the passage of new laws [118]. Additionally, other classifications based on the urgency [119], [120] and timing [121] of an engineering change have been proposed.

Jarratt et al. also advocate for the resolution of engineering changes as quickly as possible to reduce the cost of implementation. The Rule of 10, as reported by [122], [123], estimates that the expense of implementation increases, on average, by a factor of 10 as each phase of the design process passes. Therefore, it is important to address changes earlier in the design process so that costs can be minimized. Additionally, as the design process is carried out, more stakeholders may become involved. This strains the lines of communication because information must be shared among more people. Because of this, some parties may be lacking information during times where engineering changes ought to be implemented, leading to delays and extra cost [116], [117]. It is for these reasons, and the couplings [124] of components and manufacturing, components within a subsystem, and components in different subsystems, that can cause engineering change to propagate.

To manage engineering change and avoid engineering change propagation, CPM is used to integrate two DSMs to analyze the risk of change by taking into account the likelihood and impact of that change [125]. Clarkson et al. [2] use DSMs to model ECNs and draw relationships among them to predict and address change propagation. Furthermore, Morkos [4], Hein [5], and Menon [6] adapt this approach to the prediction of change and change propagation through management of requirements.

In [4], Morkos documents the creation and use of the ARCPP tool in predicting requirement change propagation. In this work, the requirements of the Toho and Pierburg projects are first correlated with the ECNs provided with the documentation. The requirements associated with the implemented ECNs were then investigated for any relationships among them that might set them apart as being involved in a situation of change propagation. Those that were confirmed as related were validated as having changed because of propagation through contact with the managing company and the documentation they provided.
The ARCPP was leveraged in this case to determine the relationship scores based on combinations of fifteen relators assembled from nouns, verbs, and keywords identified by experts. These fifteen relators were used to find potential relationships between requirements in the form of DSMs. The three optimal DSMs, determined through the application of three filters detailed in the section on the Automated Requirement Change Propagation Prediction Tool, were then overlaid to get the total relationship scores for requirements in both Toho and Pierburg. These relationship scores were then converted into propagation scores using the RMS formula found in Equation 3. By investigating the requirements with the highest propagation scores, Morkos was able to use the tool to correctly determine the requirements that actually changed in each project.

In [5], Hein explores the influence of physical (nouns) and functional (verbs) domains on the prediction of change propagation. In his analysis, he varies the number of nouns and verbs used by the ARCPP to determine the relationship scores and the subsequent propagation scores. Hein determined that the physical domain had more influence on the prediction of change and change propagation and that, while the functional domain was able to provide some resolution in conjunction with the physical domain, change propagation could be predicted using only nouns.

In [6], Menon investigates the ability of complex network metrics for the prediction of change propagation within requirements documents. In his analysis, he successfully applies graph theoretic metrics to the requirement networks and is able to predict the changes. However, he notes in his conclusion that the prediction capabilities change based on the network models used to represent requirements since every requirements document is unique and the analysis must be tailored to each representation.

Presented in this thesis is a combination of the previously discussed topics, where engineering change and change propagation is addressed through the management of requirements. Furthermore, the author seeks to broaden the understanding of change and change propagation by investigating from the perspectives of requirements after separating them into functional and nonfunctional types. Addressing change and change propagation
using requirement functionality can enable engineers and designers to leverage the current body of knowledge regarding each type of requirement to also manage engineering change and change propagation. Functional requirements can still be used to tag engineering knowledge and make design decisions and, a knowledge of how this type of requirement affects change and change propagation can allow engineers and designers to make more enlightened design decisions so that the impact of change and change propagation on project budgets and timelines can be minimized. Furthermore, a knowledge of nonfunctional requirements and their impact on change and change propagation may enable engineers and designers to rank their importance more precisely and allow them to make decisions that satisfy stakeholders and other external influences while minimizing redesign efforts to other nonfunctional and functional requirements.

Functional and nonfunctional requirements separated from each other by asking the question “Does the requirement prescribe something for the project to accomplish?” They were then studied using the ARCPP tool. Using the tool’s capability to generate relationship and propagation scores, the author was able to analyze the scores to determine the strength of relationship among requirements, the likelihood of change propagation, and the number of relationships and potential propagation paths that exist with each requirement.

**Discussion of Results**

The results of this study are intended to enable engineers and designers to explore the relationships among requirements. Addressing both RQ1 and RQ2 allows conclusions about the nature of change and change propagation in requirements to be drawn and provides a recommendation to which relationships should be investigated more thoroughly to predict possible propagation. Furthermore, a broadened understanding of change and change propagation may provide a recommendation for how customer desires may be translated into functional and nonfunctional requirements to enable design teams to analyze and predict change and change propagation more easily than current methods allow.
As reported below, the conclusions for RQ1 and RQ2 are listed in Table 21 and Table 22.

**Table 21 – Addressed RQ1**

<table>
<thead>
<tr>
<th>Question</th>
<th>Hypothesis</th>
<th>Verification Method</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does the requirement type, as classified by functional or nonfunctional, influence change and change propagation?</td>
<td>Requirement type, whether functional or nonfunctional, will have some noticeable influence on the change and change propagation characteristics of each requirement.</td>
<td>Split the two sets of project requirements into functional and nonfunctional types and analyze the relationships made among requirements as well as the population distribution of each type within the requirements documents.</td>
<td>Functional and nonfunctional requirements form the strongest relationships with other requirements of the same type. Furthermore, depending on the population breakdown of a requirements document, the number of propagation relationships will skew towards the type of requirement that appears the most. Additionally, the propagation scores show that change propagation occurs through nonfunctional requirements more readily than through functional requirements.</td>
</tr>
</tbody>
</table>
Research Question 2

<table>
<thead>
<tr>
<th>Question</th>
<th>Does a requirement’s status as functional or nonfunctional make it more likely to propagate change forward to others of the same type?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypothesis</td>
<td>There is a possibility that requirements may propagate change across functional boundaries; however, this is counterintuitive because most researchers would expect nonfunctional requirements to affect only other nonfunctional requirements, and likewise for functional requirements.</td>
</tr>
<tr>
<td>Verification Method</td>
<td>Examine the relationship and propagation scores among requirements and run statistical tests to determine the strength of relationship between the different requirement types.</td>
</tr>
<tr>
<td>Conclusion</td>
<td>Functional and nonfunctional requirements are not more inclined to propagate change within functional boundaries; however, nonfunctional requirements are more likely to instigate change, overall, than functional requirements. Furthermore, the probability of change propagation to a requirement type, driven by the number of that type in a requirement document, in combination with propagation scores, may explain the discrepancies between the actual changes in Toho and Pierburg and the changes one might expect based solely on the experimentally determined propagation scores and relationships.</td>
</tr>
</tbody>
</table>
As stated in Table 21, requirements are related to each other most strongly within functional boundaries. In regard to propagation scores, requirements scored higher when propagation was predicted to the requirement type appearing most in the requirement document. However, in all cases, the requirement instigating the strongest propagation scores was nonfunctional. In Toho, nonfunctional requirements were most likely to receive change propagation, whereas in Pierburg, functional requirements were most likely receive change propagation.

That nonfunctional requirements should be the more likely of the two types to propagate change is surprising. This indicates that externally imposed changes may be more difficult to integrate with current designs than changes proposed from within. Furthermore, it is interesting that the requirements governing such project items as ergonomics, recyclability, maintainability, and others, should influence, not only other requirements of the same type, but requirements that govern the products performance of task.

In RQ2, it is discovered that change propagation is not relegated to the boundaries of requirement functionality. Instead, it appears that change propagation may be influenced more by how many of each type of requirement are included in a requirement document. Because of this, it is important for engineers and designers to understand how functional and nonfunctional requirements are related to each other and how propagation scores are influenced by requirement type population.

Additionally, the actual propagations were investigated to see how they compared to the experimental results. Upon examination, it is observed that, despite the prevalence of propagation paths to nonfunctional requirements, in Toho, most propagation occurred between functional requirements. Likewise, the actual propagation of Pierburg requirements indicates that nearly equal numbers of functional and nonfunctional requirements were involved in changes. These results are perplexing because they do not match with the experimental results determined from the relationship and propagation scores generated from the ARCPP tool.
Because the results of these data sets do not agree, the author investigated the probability of each requirement type in receiving change propagation by the amount of each appearing in each project’s requirement document. In examining the probability of receiving change, it is observed that, despite numerical advantages, there was a slight bias in probability of propagation to the requirement type represented less in a requirement document at the individual change level. Because of this, the author speculates that it may be that the combination of probability scores and propagation scores provide a more accurate prediction of change and change propagation, if they can be related by some function. Unfortunately, the author regrets that there was not enough time to try to determine how this function might be discovered.

**Recommendations for Implementation**

In implementing the results of the study, the author would recommend that the determination of requirement functionality and calculation of probability scores be integrated with the ARCPP. The ARCPP can currently be used to generate both the relationship and propagation scores of each requirement; however, generating requirement functionality and probability scores may enable engineers to predict propagation paths from the perspective of requirement functionality, as well as the predictive capabilities the ARCPP already offers, and afford them the ability to describe the changes with more specificity.

For example, if it is determined that a nonfunctional requirement needs to change because of new legislation passed, then knowing that nonfunctional requirements may be the most likely to propagate change is helpful. Furthermore, understanding how probable change is to propagate to either type, based on how many of each type appear in a requirement document, may refine any initial estimations of the nonfunctional requirement’s impact. The design team could then take the nonfunctional requirement under advisement and investigate, first, that requirement’s relationship to other requirements (based on the population of each requirement type in the requirement document) and consider whether these are related closely enough to receive change. The next step would be to investigate
the potential propagation paths. Furthermore, even if the propagation paths and scores are not yet known, the design team should recall that a greater number of propagation paths will exist between the nonfunctional requirement being examined and whatever type of requirement appears the most in the requirement document.

Conversely, if a functional requirement is being considered for change, then the engineers and designers of a should consider its relationships with other functional requirements first because there is a greater chance that if it changes, one of them may change. After considering along this relationship path, the engineers and designers may investigate the other potential propagation paths based on the populations of each requirement type in the requirement document.

Furthermore, understanding how requirement functionality affects change and change propagation may help engineers when they elicit requirements from external sources. Further research is required; however, understanding that nonfunctional requirements may form the strongest relationships that may instigate change with other requirements helps engineers and designers in requesting requirements. NonFunctional requirements may be requested with more specificity so that the relationships they form are both stronger and potentially less widespread. Additionally, greater consideration can be given to nonfunctional requirements when prioritizing stakeholder desires. Finally, design teams may choose to disregard externally requested requirements after the initial elicitation phase so that the influence of nonfunctional requirements on the project may be minimized. Of course, this does not apply to regulatory changes; however, stakeholder desires may be considered static so that changes to nonfunctional requirements are minimized.

Finally, this research provides value to the ARCPP because it demonstrates that the data produced can not only be used to predict potential change and change propagations using requirement documents, but can also be used to characterize requirements themselves and provide data detailing the potential behaviors of the different types of requirements. Furthermore, the calculations and spreadsheet manipulations can be integrated into the ARCPP so that the entire process can be scaled to other projects of greater size and scope.
Therefore, the ARCPP can potentially be utilized for further research regarding the different behaviors of functional and nonfunctional requirements and may be leveraged for determining whether there are different degrees of functionality and their impacts and whether functional and nonfunctional requirements at a subsystem level will behave similarly at the system level, and vice versa.

**Future Work**

This thesis presents a preliminary estimation of how requirement type may influence change and change propagation and how the ARCPP and the results it generates might be leveraged to estimate the behaviors of each type. Requirements are most connected with others of the same type. However, NonFunctional requirements are most likely to instigate change propagation, which also means that, in cases where functional requirements outnumber nonfunctional requirements, higher relationship scores are reported from an instigating nonfunctional requirement to a functional requirement receiving change; this is the only case that is contrary to the assertion that relationship strength is greatest within the boundaries of functionality. Furthermore, it was discovered that change propagation is less dependent on requirement functionality than it is dependent on the functional characteristics of the requirements document. That is, how many of each type of requirement are represented in the document.

As mentioned previously, the Toho and Pierburg projects did not behave as expected. Despite the indications that nonfunctional requirements were most likely to be involved in change and change propagation in Toho, functional requirements were more involved in the actual case. Likewise, the data indicated that the functional requirements of the Pierburg project were almost twice as likely to be involved in change and change propagation, yet there was an even number of functional and nonfunctional requirements involved in the actual changes. Further research may be performed on the Toho and Pierburg projects to investigate why they did not behave according to the experimentally indicated results.
Additionally, because the data indicates that change propagation is dependent on how many of each type of requirement appears in the requirements document, the author would recommend using further projects for investigation into how the quantities of functional and nonfunctional requirements affect change propagation paths.

Finally, research into the different impacts of emergent and initiated changes on change propagation may be of value because initiated changes may affect nonfunctional requirements or design team generated requirements of both types, whereas emergent changes may affect more functional requirements than nonfunctional requirements.

Because there is further research to be performed on the impact of functional and nonfunctional requirements and emergent versus initiated changes on change and change propagation, the following research questions posed in Table 23 and Table 24 can be considered for future work.
Table 23 - Proposed Research Questions for Future Work: Part 1

<table>
<thead>
<tr>
<th>Future Research 1</th>
<th>Question</th>
<th>Do change propagation paths vary based on the quantity of each type that appears in requirement documents?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypothesis</td>
<td></td>
<td>Change propagation paths will vary to functional or nonfunctional requirements based on the quantity of each that appear in requirement documents.</td>
</tr>
<tr>
<td>Verification</td>
<td>Method</td>
<td>Procure additional sets of requirements for analysis and compare results of propagation paths to quantities of each type of requirement.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Future Research 2</th>
<th>Question</th>
<th>Can relationship strength be correlated with the quantity of each type of requirement that appears in requirement documents?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypothesis</td>
<td></td>
<td>Relationship strength can be correlated with the quantity of each type of requirement that appears in requirement documents.</td>
</tr>
<tr>
<td>Verification</td>
<td>Method</td>
<td>Procure additional sets of requirements for analysis and compare results of relationship strengths to quantities of each type of requirement.</td>
</tr>
</tbody>
</table>
Table 24 - Proposed Research Questions for Future Work: Part 2

<table>
<thead>
<tr>
<th>Future Research 3</th>
<th>Question</th>
<th>Do specific sub-classifications of functional and nonfunctional requirements impact change and change propagation differently?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hypothesis</td>
<td>Specific sub-classifications of functional and nonfunctional requirements impact change and change propagation differently from each other.</td>
</tr>
<tr>
<td></td>
<td>Verification Method</td>
<td>Separate functional and nonfunctional requirements into further sub-classifications to determine the influence of these types on change and change propagation.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Future Research 4</th>
<th>Question</th>
<th>Does the type of change, emergent or initiated, impact change and change propagation through interaction with functional and nonfunctional requirements?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hypothesis</td>
<td>There may be differences in how the two types of change affect change and change propagation through interaction with functional and nonfunctional requirements.</td>
</tr>
<tr>
<td></td>
<td>Verification Method</td>
<td>Investigate ECNs for their type, whether emergent or initiated, and correlate to the change propagation characteristics of functional and nonfunctional requirements as generated by the ARCPP and method presented here.</td>
</tr>
</tbody>
</table>

The future research questions presented in Table 23 and Table 24 can be explored in future research projects to further broaden researchers’ understanding of change and change propagation. Future Research 1 (FR1), Future Research 2 (FR2), Future Research 3 (FR3), and Future Research 4 (FR4) if addressed, will provide insight into the more ambiguous conclusions drawn from the research presented in this thesis. In particular, FR1 addresses whether propagation paths are confirmed as being influenced by the quantity of each type.
of requirement appearing in requirement documents. Additionally, FR2 correlates the strength of relationships formed with the quantity of each requirement type that appears in requirement documents. FR3 adds further resolution to the understanding of change and change propagation by subdividing each type of requirement into more discrete entities. If addressing this third question, in particular, is successful, specific elements of requirements in requirement documents may be recommended for examination for potential change and change propagation relationships. Finally, FR4, enables engineers an additional method of evaluating the behavior of change and change propagation through requirement documents based on the originator of a change. This would enable design teams to assess the impact of changing a requirement as the requirements are written. Addressing these three future research questions broaden researchers’ understanding of change and change propagation and may further assist design teams in managing engineering change.
References


