Analyzing the United States Air Force Agile Transformation
Using a Systems Thinking Approach

by

Elizabeth A. Bieler

B.S. Electrical Engineering (2013)
Miami University of Ohio
Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degrees of
Master of Science in Engineering and Management
at the
Massachusetts Institute of Technology
September 2018

© 2018 Elizabeth A. Bieler.
All rights reserved.

The author hereby grants to MIT permissions to reproduce and to distribute publicly paper and electronic copies of
this thesis document in whole or in part in any medium now known or hereafter created.
Analyzing the United States Air Force Agile Transformation Using a Systems Thinking Approach

by

Elizabeth A. Bieler

Submitted to the Department of System Design and Management on Aug 10, 2018 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Engineering and Management

ABSTRACT

The United States Air Force (USAF), like many large enterprises, has struggled to keep up with industry in developing state of the art systems in a timely manner. Although many argue that the landscape for the Department of Defense (DoD) has complex and unique factors this does not make the needs of the warfighter to be less urgent. To improve its current state, the USAF has been trying to transform software acquisitions from a Waterfall to Agile approach. Progress towards an Agile transformation appears to be somewhat stagnant.

This thesis analyzes the status of the USAF Agile transformation; evaluates challenges for the USAF in becoming truly Agile - and proposes recommendations to the USAF, DoD and government leaders on how to make the USAF Agile transformation successful.

A main finding of this investigation is that the USAF Agile transformation is experiencing the same challenges felt by many large corporations; however, they have a very unique set of stakeholders. The current approach needs modification to include these stakeholders’ needs. There are actions that can be taken by personnel at very different levels to make the Agile transformation successful. Challenges such as regulations that reflect Waterfall methodologies are best addressed by high level leadership while others are best addressed by employees such as creating a grassroots movement.

Agile could be a major contributor for helping the USAF to succeed at becoming more effective at quickly deploying needed systems. By utilizing a holistic view, the USAF will be better able to address the current challenges of Agile transformation to deliver capability to the warfighter.

Thesis Advisor: Dr. Donna H. Rhodes
Title: Principal Research Scientist, Sociotechnical Systems Research Center
Acknowledgements

This thesis has been a great learning journey for me, guided and supported by my advisor: Dr. Donna Rhodes. She helped me work through the many challenges intellectual, bureaucratic and personal. Her enthusiasm, encouragement and optimism made completing this thesis possible. This thesis also would not have been possible without the financial funding for my school by the USAF Science and Engineering PALACE Acquire Program.

Thank you to all of the engineers and program managers who work for the USAF and made time to discuss your experiences with me. Although I cannot list your names – I learned the most from our time together and your candor.

Sincere appreciation goes to my family and friends who have been not only been understanding but have cheered me on with enthusiasm. To my mother, Deborah Ahles, for fostering a love of science at the youngest possible age and leading by example. She demonstrated that learning is a lifelong process. To my father, Mark Ahles, for helping me find a love for calculus and physics. Without your encouragement I may have never considered becoming an engineer. To my sisters and brother, Rachel Tritle, Sarah Ahles and Aaron Ahles, for exploring the world through science with me. To my mother-in-law, Kathryn Bieler, who has mentored me as one of her own offering great insights into technical leadership and business. To my father-in-law, Jeffery Bieler, for his unfailing emotional support. To my sister-in-law, Kimberly Cahill, for her heart-warming kindness.

Most importantly, I thank with love, my husband Gregory Bieler. He believed in me before I believed in myself. He has been understanding and supportive of this challenging time showing me unconditional love.
Dedication

To my family and friends, without your support, I would have none of my successes.

To all civil servants, the moment we stop pushing for better is the one in which bureaucracy wins.
# Table of Contents

ABSTRACT .............................................................................................................................. 3  
ACKNOWLEDGEMENTS ........................................................................................................ 4  
DEDICATION .......................................................................................................................... 5  
TABLE OF CONTENTS ........................................................................................................... 7  
LIST OF FIGURES ................................................................................................................ 10  
LIST OF TABLES .................................................................................................................. 11  

1 INTRODUCTION ................................................................................................................ 13  
   1.1 MOTIVATION ................................................................................................................. 13  
      1.1.1 Establishing a Need for a New Methodology in the USAF ...................................... 14  
      1.1.2 Establishing a Need for a Systems View of USAF Acquisitions .............................. 15  
   1.2 THESIS OBJECTIVES .................................................................................................... 16  
   1.3 RESEARCH METHODS ................................................................................................. 17  
      1.3.1 Expert Feedback ..................................................................................................... 18  
   1.4 THESIS STRUCTURE ..................................................................................................... 19  

2 DEFINING A NEED FOR AGILE ....................................................................................... 20  
   2.1 DEFINING AGILE ......................................................................................................... 20  
      2.1.1 The Plan-Do-Check-Act Cycle .............................................................................. 21  
      2.1.1.1 Agile DevOps ................................................................................................ 23  
      2.1.1.2 Other Types of Agile .................................................................................. 24  
      2.1.2 The Foundation of an Agile Project ...................................................................... 24  
      2.1.3 Agile Company Culture ..................................................................................... 26  
      2.1.4 The Pros and Cons of an Agile Project ................................................................ 27  
      2.1.4.1 The Advantages of an Agile Project ............................................................ 27  
      2.1.4.2 The Disadvantages of an Agile Project ........................................................ 28  
      2.1.5 Defining Large-Scale Agile .................................................................................. 29  
      2.1.6 Agile in Regulated and Safety Critical Environments .......................................... 31  
         2.1.6.1 National Security Agency ............................................................................ 32  
         2.1.6.2 SpaceX ........................................................................................................... 32  
   2.2 DEFINING AN AGILE TEAM ....................................................................................... 32  
      2.2.1 Agile Team Makeup ............................................................................................ 33  
      2.2.2 Company Support for an Agile Team .................................................................. 34  
      2.2.3 Agile Team Member Training ............................................................................. 35  
      2.2.4 Customer and Agile Team Relationship .............................................................. 35  
   2.3 AGILE PRODUCT REQUIREMENTS, TESTING AND IMPLEMENTATION ............. 36  
      2.3.1 How Requirements are Developed .................................................................... 36  
      2.3.2 When Requirements are Developed ................................................................... 37  
      2.3.3 How are Products Tested .................................................................................... 37  
      2.3.4 When are Products Implemented ...................................................................... 38  
      2.3.5 Agile vs. Waterfall Product Requirements, Testing and Implementation ............ 39
List of Figures

Figure 1.1. Research Approach ............................................................................................................. 18
Figure 2.1 PDCA Cycle adapted from [15] .......................................................................................... 21
Figure 2.2 Waterfall PDCA Cycle adapted from [3], [12] .................................................................. 21
Figure 2.3 Agile PDCA Cycle adapted from [3], [12], [13] ................................................................. 22
Figure 2.4 Agile vs. Agile DevOps Sourced [3] ................................................................................... 24
Figure 2.5 Deployment and Integration Team Makeup Sourced [12] ................................................... 34
Figure 3.1 Lenses for Different Views of an Enterprise Sourced [8] .................................................... 41
Figure 3.2 Model of Commercial Software Adaption Sourced [27] .................................................... 45
Figure 3.3 Model of a Mainly Software Defense Program Sourced [27] ............................................. 46
Figure 3.4 Stakeholder Map for Acquisition Program .......................................................................... 52
Figure 3.5 High-Level Chain of Command Adapted from [27] .......................................................... 57
List of Tables

Table 1-1. Thesis Structure ........................................................................................................ 19
Table 2-1 Traditional Waterfall vs. Agile Source [20] .......................................................... 39
Table 3-1 Summary of Ecosystem Factor Analysis ................................................................ 43
Table 4-1 Summary of Challenges ......................................................................................... 68
Table 4-2 Summary of Challenge Mitigations ....................................................................... 77
Table 4-3 Summary of SWOT Analysis .................................................................................. 80
1 Introduction

1.1 Motivation

To many in the defense community the book Augustine’s Laws by Norman Augustine is infamous for its satirical commentary on the challenges of being a defense contractor. For example, he found that annual cost of the defense budget is not growing nearly as quickly as the cost to develop aircraft resulting in the following law [1].

“Law XVI: In the year 2054, the entire defense budget will purchase just one aircraft. This aircraft will have to be shared by the Air Force and Navy \( \frac{3}{2} \) days each per week except for leap year, when it will be made available to the marines for the extra day.” [1]

Although this book was published in 1984, many of the problems mentioned are still relevant today[1]. Many of them seem to be addressed by agile methodologies with promises of either a shortened development, better products or less cost [2]–[5].

Agile principles are not new; they have been around for nearly 20 years. While DevOps is newer, many of the key principles and values are the same. These concepts were at first mainly embraced by small startups whose company cultures aligned well with agile principles. As the success stories of agile companies and products grew, so did interest from larger companies. Success stories of better products and shorter program lengths have led to commercial giants and even the Department of Defense (DoD) to become interested. It seemed this could not come at better time considering the mostly retracting defense budgets with lengthy, expensive development of weapon systems. Despite this the defense industry is struggling to figure out how to implement agile methodologies. The United States Air Force (USAF) is no exception to this struggle.

In some ways the United States Air Force is no different than many large established commercial companies. Although there are some well-known large companies, such as Netflix and Amazon, that utilize Agile or DevOps these companies are often software heavy. In contrast, other larger companies have struggled with how to implement Agile especially in an effective way. However, there are also many unique aspects to the DoD since it is not commercial but a
government entity. Being a government entity effects many areas of the development of weapon systems. One example is the involvement of Congress who has many roles, but one of the most noteworthy is in the oversight on acquisition programs. Considering the uniqueness of the DoD it is important to utilize a holistic view to understand all the challenges of implementation.

It is important to understand the challenges being tackled as the USAF, and the DoD, continue to integrate Agile. In order to have a successful implementation, the problems need to be well understood from a holistic perspective. This thesis identifies the challenges being experienced by the USAF, and brings a system view to help our leaders address pressing issues so the USAF is able to successfully implement Agile throughout the USAF. This hopefully saves tax dollars and time while delivering the best quality weapon systems to the field to protect not only the military members but the American people.

1.1.1 Establishing a Need for a New Methodology in the USAF

Adversaries to the United States (US) are obtaining capabilities at unprecedented speeds [3]. Weapon systems are growing increasingly complex [6]. Technical needs often shift over time requiring expensive changes leading to cost and schedule overruns [6]. At the same time the percentage of software of a weapon system is growing at an unparalleled pace and is 60% of the risk for the average program [3]. This leads to new challenges compared to mainly hardware systems. Compared to hardware, software leads to breakdowns and has unique requirements for security, testing and maintenance [3]. In further contrast with hardware, software never needs to be replaced from wear and tear, instead staying with the weapon system until death [3]. Exacerbating the problem is the use of unvetted technologies that are often a key component, greatly increasing technical risk [6]. Unfortunately, technical challenges are far from the only noted problem.

Within the DoD, cost estimates for a product are determined before even knowing if it’s technically feasible, so estimates are always wrong and all stakeholders know it [6]. During an investigation by the Defense Science Board, DSB, they found that both cost and schedule overruns were likely due to software that could have been avoided by using Agile [3]. Both colossal cost and schedule overruns are common in acquisition programs [6]. The cost overruns are so persistent that the average program is 21 months behind schedule, with the worst programs
being almost ten years late [6]. Despite cost overruns, huge technical obstacles, amongst other problems over 95% of acquisition programs are never canceled [6]. Defense projects have continued to utilize securely controlled procedures even though cost and schedule are pointing to catastrophe [5]. Perhaps this is because success for an acquisition program is measured by getting funding, not for delivering a weapon system to the war fighter [6].

There is a growing clamoring for a need for change within acquisitions. Ret. Army Gen. Hirsch spoke about how the acquisition processes set people up to fail [6]. The current procedures seem to mitigate risk so that leaders who adhere to them are not responsible for failures [6]. Top military leaders are concerned about what appears to be exponentially growing costs [6]. Failing procedures, seemingly ever-growing cost and schedules along with complex technical challenges are all signs indicating that the current system is failing to deliver. The current path that USAF acquisitions is on is unsustainable. Better solutions are needed to meet the pressing needs so the US military can protect and defend for a lesser cost and more credible schedule.

1.1.2 Establishing a Need for a Systems View of USAF Acquisitions

The USAF is part of a complex and complicated environment with individuals, companies and even other countries that all have a stake in the results of USAF acquisition projects and programs. For example, stakeholders in USAF acquisitions include congressmen, contractors, lobbyists, service members, congressional constituents to name a few [6]. All of these stakeholders put immense pressure resulting in programs that rarely, if ever, are canceled [6]. Even within the DoD there are lots of rewards for low estimates for cost and extreme optimism for technical hurdles [6]. Because it is such a complicated and complex system it demands holistic analysis.

Holistic thinking involves thinking from different perspectives and examining anything that might interact with the system [7]. One way to approach holistic thinking is by looking at the system through multiple views to avoid concentration on only one area [8].

“Holism insists on the intimate interconnection of things–on the idea of the whole–and to think holistically is to think deliberately about the whole.” [7]
In order to solve today’s business problems, we need to focus on more than the immediate task at hand [9]. For businesses, systems thinking can be the difference between success and failure [10]. Oversights and even disasters tend to happen when the focus is concentrated on one aspect [8]. Instead we should consider the context to come up with truly revolutionary solutions [9].

Holistic thinking helps identify as many as possible unknown unknowns, which can be the most detrimental to a system [7]. An unknown unknown is being on vacation at the beach completely unaware that a hurricane is heading towards you and will hit land soon [7]. A known unknown is knowing the hurricane is heading towards your beach vacation, but you do not know when it is going to hit or the category [7]. Although both are detrimental to your vacation, only the unknown unknown is also likely detrimental to your life [7].

Holistic thinking is associated with systems thinking. Systems thinking is beneficial because of systems inherit qualities. There are very few things that cannot be categorized as a system [7]. A key attribute of a system is that when the individual components work together the result is a new function different than the individual components – this is called emergence [7]. This emergence is a critical trait; it gives both the beauty (helpful emergence) and danger (undesired emergence) of a system [7].

Because of the immense values of analyzing problems using holistic systems thinking it is not only helpful but necessary to truly analyze the challenges with implementing Agile in the USAF. Failing to do so misses critical pieces of the puzzle ultimately leading to misleading conclusions.

1.2 Thesis Objectives

The aim of this thesis began with a research question: Is there a need for the USAF to adapt Agile methodologies? This first research question provided motivation to understand Agile and its fundamental differences with Waterfall methodologies. Since there is an established need for a new methodology this provided motivation to determine if Agile would be able to address those problems.

This led to a second research question: What challenges will the USAF face as it transforms to be an Agile organization? In other words, an investigation of the challenges that will be faced as
an extremely large organization tries to adapt the Agile principals that work well for small companies. In addition, it motivates an analysis of what unique challenges the USAF may face as both a government organization and a military one.

The third research question then is: *In what ways can these challenges be mitigated?* The goal of this question is to use the thorough analysis of both Agile methods and the holistic view of United States Air Force to find ways to address these challenges.

### 1.3 Research Methods

Figure 1.1 shows the continuous iterative process used to perform this research. This process always came back to and began with knowledge gathering, primarily through literature review. As knowledge was gathered, I formulated theories to be tested by further knowledge gathering through both literature and subject-matter experts. As theories matured it led to the simplified model to capture the key characteristics of the “system” of USAF Acquisitions. The term model is being used to mean a simplified representation of a real system. Simplified representations can add value by making a complex system easier for human comprehension to understand. This model went through further internal validation through literature review, with refinement of theories. An exploratory approach was taken to generate a holistic view. As new information was discovered, theories and the model were updated. Eventually, this led to a review by USAF Agile experts, with years of experience in the USAF and experience on an Agile program. Expert feedback, along with further research, theory and model refinement led to a finalized model.
Continuous Iterative Process:

1. Knowledge Gathering (literature review)
2. Formulation of theories
3. Generating of models
4. Review by experts
5. Finalized models

Figure 1.1. Research Approach

1.3.1 Expert Feedback

Expert feedback was gathered through interviews in which the models (Table 3-1, Figure 3.4, Table 4-1, Table 4-2, Table 4-3) presented in this thesis were reviewed and discussed. This discussion was utilized to determine if the models matched or differed from their experience with Agile in the Air Force. These interviews were kept anonymous to encourage honesty about their experience without fear of retribution, whether positive or negative. In total five individuals were interviewed from four different Agile programs at varying levels of experience and different level of roles. Expert interviewees had at least five years of experience in the USAF (most had more) and at least one year of USAF Agile experience with the exception of one interviewee who had three years of USAF time.

Using personal experience of five individuals to validate or invalidate is an imperfect approach. Data could be skewed with such a small sample size or geographic homogeneity since all were assigned to the same Air Force Base. Due to the small sample size outlier perspectives were excluded from the central analysis since they seemed related to personnel experience rather than reflective of the great picture. However, since each interview was conducted individually with no knowledge of who the other interviewees are, there is less chance for contamination between interviews. Also, some of the answers interviewees gave conflicted to some extent with each other. For example, one interviewee felt that the major command requesting the weapon system
was a large influence on the Agile process and should be included as a stakeholder separate from the warfighter. Another interviewee felt that for the level of the diagram that this was inappropriate and saw the major command and the warfighter as the same stakeholder. For this reason and others, it is important to note the limitations of interviewee sample size in this research. Certainly, further research would warrant further confidence with a more rigorous approach. Unfortunately, time limits prevented more extensive data collection for this thesis, and therefore findings should be considered as an initial exploration of the topic.

1.4 Thesis Structure

This thesis is arranged as follows, and is summarized in Table 1-1. Chapter 2 establishes the need for Agile methodologies in the USAF by investigating how the current problems in USAF acquisitions can be addressed by Agile but not Waterfall (the current) methodologies. Chapter 3 lays out the current state of the USAF Agile transformation that is in progress. Chapter 4 synthesizes the challenges of Agile transformation and presents the framework for modeling the system. Chapter 5 explores mitigation actions that could be taken to address the challenges. Chapter 6 summarizes the thesis findings and prospective future work.

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ch. 1</td>
<td>Introduction and Research Methods</td>
</tr>
<tr>
<td>Ch. 2</td>
<td>Defining a Need for Agile and DevOps</td>
</tr>
<tr>
<td>Ch. 3</td>
<td>Current Status of Agile in the USAF</td>
</tr>
<tr>
<td>Ch. 4</td>
<td>Synthesis of Challenges in the Model</td>
</tr>
<tr>
<td>Ch. 5</td>
<td>Mitigation Actions for Challenges</td>
</tr>
<tr>
<td>Ch. 6</td>
<td>Conclusions and Future Work</td>
</tr>
</tbody>
</table>

Table 1-1. Thesis Structure
2 Defining a Need for Agile

If only looking at the headlines Agile appears to be the new miracle drug for projects – a long list of benefits with zero side effects. For example the Harvard Business Review states that Agile transformed software by increasing speed, quality and productivity [11]. With a little initial investigative research an Agile novice may become even more confused by a seemingly endless proliferation of jargon unique only to Agile. This makes it challenging to assess if Agile is a good solution for a project or company’s needs. This chapter clarifies what Agile is, why it is a good fit for addressing the challenges in today’s USAF, and explains how Agile can work for large scale enterprises.

2.1 Defining Agile

Our world has become digital [12]. For many companies software is not only the product but also what differentiates their company [12]. Nearly 80% of companies state that delivery within 2 months is too slow [12]. Because of this 75% of companies have moved to Agile or Agile variation to increase delivery rates [12]. In Dioquino’s presentation he defines Agile as a way to increase delivery with better feedback from customers and a focus on ROI [12]. According to Serrador and Pinto, Agile is iterative and incremental with fluid requirements and scope, and high customer engagement [13]. In the Harvard Business Review article the authors chose to focus on Agile scrum defining it as empowering a cross-functional small team that manages itself and is responsible for all work [11]. Rico defines Agile as a method of software design that utilizes small teams to iteratively advance products through customer feedback [14]. With these definitions in mind, the following section reviews methodologies that pre-date Agile, working forward to modern methodology.
2.1.1 The Plan-Do-Check-Act Cycle

The Plan-Do-Check-Act (PDCA) cycle, shown below (Figure 2.1), was first created in the 1930’s by Dr. Shewhart [15]. This was considered fundamental in determining when action was required versus leaving it be [15].

Figure 2.1 PDCA Cycle adapted from [15]

This simple, but elegant method is helpful in explaining some of the key differences between Agile and Waterfall philosophies. It can even clarify some of the challenges inherit in Waterfall for software projects. It is nearly impossible to discuss Agile without discussing Waterfall, whose short falls, particularly in the software realm are credited with creating the need for a new process [16]. Waterfall could be described as one long PDCA cycle. First all requirements for project our written, second it is designed, third code is written, then the software is tested and then finally fielded [3], [16].

Figure 2.2 Waterfall PDCA Cycle adapted from [3], [12]
In contrast Agile takes a small functional piece of a project and designs, writes, tests and sometimes fields the software – this is called an iteration [3], [16], [17]. These iterations are repeated until the entire project is completed [3], [16], [17].

Figure 2.3 Agile PDCA Cycle adapted from [3], [12], [13]

Waterfall was and is embraced for seeming to organize programming by giving control and predictability [3], [5], [17]. The main value that Waterfall methods are found to deliver is quality documentation and processes [14]. Processes that required change authorizations and gateways to proceed to the next step confirm feelings of control [2], [3], [17]. The DoD embraced these methods as risk reducing, up to the year 2000 it followed a modified Waterfall process [3]. DoD requirements were finalized, cost and schedule estimates were made (and approved) then contracting sent out a request for proposal [3]. Then the software was developed (the number of members is based on a number of lines of code estimate) and tested before being fielded [3]. Waterfall was seen as the best way to ensure maintainability, reliability and safety [14]. Despite these processes Waterfall failed at reducing software risk [3], [13].

One of the key reasons why Waterfall tends to fail in comparison to Agile is in the delay of the PDCA loop [2] - [3]. Errors are not found in requirements until (on a large project) years later during testing or fielding when users interact with the product [3]. By this point changes to the
project are extremely expensive to make and will likely delay the project [2], [16], [17]. Choices are then made between adding value or meeting program management goals [2], [16], [17]. This also results in inordinate amounts of rework, client disappointment, and the risk that once the project is complete the current technology has superseded it [13]. Waterfall assumes that all requirements are fully understood at a detailed level [2], [16], [17]. When errors are found in a Waterfall project the plan often takes priority [5]. Not surprisingly change has been found to be inescapable during project planning [13]. Agile distinguishes itself from Waterfall by avoiding these faulty assumptions.

Agile’s use of “continuous iterative” development does not waste time on writing lengthy requirements documents [3], [16]. Writing unnecessary documentation adds to Waterfall projects rigidity [13], [5]. Instead Agile processes use broad desired functions to guide design along with frequent, often daily, interactions with the user [3]. This avoids the delays in information experienced in Waterfall. Agile is also more adept at handling the inevitable challenges of designing a new product [17]. These challenges are often most effectively mitigated by increasing feedback and often results in a hardy robust product [3], [17]. As Agile has gained notoriety, many different derivatives of Agile have been developed and implemented.

2.1.1.1 Agile DevOps

Agile and DevOps are two different methodologies that have come together to create Agile DevOps. DevOps came from integrating team members from IT operations with the development team [18]. DevOps integrates more of what happens after software is fielded as shown in the Figure 2.4 [18]. Agile DevOps then involves several Agile projects at the same time for each iteration [3]. This requires careful architectural decisions and interface management [3]. There are many reasons that this method is being deployed – it can lead to quicker, more secure solutions and has heavy emphasis on automation which increases efficiency [18]. Agile DevOps is a continuous method of planning, testing, delivering and operating [12]. In order to make a method such as Agile DevOps work it requires greater automation but it integrates the whole team [12]. The increased automation also makes it easier to track key performance metrics such as quality through metrics such as number of incidents and defects or
deployment to product success rate [12]. Throughout the rest of this thesis, the term Agile is used for all Agile, Agile DevOps, DevOps and other Agile derivatives.

Figure 2.4 Agile vs. Agile DevOps Source [3]

2.1.1.2 Other Types of Agile

Unlike Agile DevOps which combined two different philosophies, there are different manifestations of Agile processes. There is a large variety of different types of Agile implementation but two of the most common are Scrum and XP, eXtreme Programming [2], [19]. Scrum, or a variation of it, is utilized five times more often than any other Agile technique [11]. However, XP, is given credit as the first Agile method and was created by Kent Beck [14]. Scrum stresses the program management side of Agile while XP stresses the increments [2], [19]. Although both emphasize the customer they do so in different ways. XP emphasizes quick, active feedback during increments that result in fast releases while Scrum emphasizes code delivery based on customer prioritized requirements [19]. Scrum also focuses on complex problem solving [11]. In addition to Scrum and XP there is also lean development that removes all waste [11]. Or there is Kanban which minimizes lead times [11]. Actual implementation of Agile often is not a pure type of Agile but rather it is a mix [2].

2.1.2 The Foundation of an Agile Project

There is a lot more to Agile then variations on series of short iterations. The Agile process begins with a list of features called a backlog that come from the customer [3]. Code is developed to meet a specific function [3]. At night, automatic testing is run on code written that
Any errors found in the automatic testing are fixed before new coding is begun the next day [3]. Once an iteration’s coding is complete it is peer reviewed, further tested, and is completed and stored in a repository [3]. Iteration(s) become a build which is automatically documented and users are able to provide feedback [3]. The feedback adds to the backlog of features [3]. The average time it takes to finish one iteration is often used as a method to track progress of an Agile team [17]. It may take a few iterations to determine this – but then burn-down charts can be created to show development [17].

Agile, however, is not only a change in procedures but also a cultural shift. The Agile Manifesto, the semi-birth place of Agile lays out four core values [16].

“(1) Individuals and interactions over processes and tools
(2) Working software over comprehensive documentation
(3) Customer collaboration over contract negotiation
(4) Responding to change over following a plan” [16].

Agile focuses on some key principals that make it substantially different then Waterfall processes. The central theme to Agile is efficient delivery of worth to the client [20]. Considering what the customer wants is not a new concept but the methodology behind the principle is. It is notoriously difficult for customers to know, let alone articulate what they want [5], [17]. Agile has a novel way to tackle this problem that is more effective [17]. The other key principle for Agile is a novel concept – embracing change rather than resisting it [17]. In order to embrace change many of the “standard” processes no longer apply. These two key principles get the heart of Agile.

“Agile software development processes can be thought of as creating options that allow decisions to be delayed until the customer needs are more clearly understood and evolving technologies have had time to mature.” [17]

This is essential in environments in which requirements are instable and the problems are convoluted [4]. That is why Agile is known for handling change well [2], [16]-[17]. Agile can do all this by enabling designers to have several options for a module where the prosperous ones are integrated [3]. Sometimes there are fears that options will lead to a never ending or never
converging project [17]. Actually, options empower the selection of a good architecture and other good selections by deferring until the last possible moment when the best data is available [3], [17]. For communication this means information is distributed as soon as it is realized even if it is incomplete [17]. Whenever possible Agile dictates that communication should be in-person not through documentation [17]. To encourage delaying judgement it is useful to talk about constraints rather than options [3], [17]. Deferring prevents a potential catastrophic error of picking the wrong choice when inefficient information is available [3].

Agile also means a shift in culture for leadership – they must learn to identify waste [17]. For example, documentation is a common area that formerly Waterfall leadership struggles [17]. Documentation should add value to the customer, which is seldom the case [17]. If it is required, such as with safety software, documentation should not just be completed but be helpful [17]. The best way to know if it is helpful is by observing to see if it is regularly used and updated without prompting [17]. Another example of waste is waiting, which is often the case when authorizations or customer feedback is needed [17]. Defects are a critical area of waste, not only in the defected code but also the time that it took to find the defect [17]. Some waste is inherently eliminated in the Agile processes, other wastes may be more easily overlooked. To identify waste, a new company culture must be developed.

2.1.3 Agile Company Culture

Limited requirements and planning aligns well with Agile principles [19]. This leads to minimized controls for upper layers but is great for developer morale [4]. One of the biggest distinctions about an Agile culture is that it empowers people [17]. It empowers them to choose the best choice from the best options [17]. In a Waterfall culture employees are told to meet cost, schedule and scope – this means they have no options from which to pick [17]. This is reflected in Dikert et al. research that one of the important success factors for large scale Agile transformations is a focus on mentality alignment [2]. They also found that management support and employee education and guidance were critical to success [2]. One example of how empowering employees to make wise decisions is creating a simple model with basic fiscal data [17]. This is giving the employees a holistic view of the problem space, rather than keeping them in silos [17]. This contrasts with Waterfall which utilizes part optimization which rarely leads to an optimized system [17]. Rather the result is often that the system is not optimized at
all [17]. This is also true for employees. Employees will not prosper when companies favor plans, procedures and documentation over everything [17].

Although most large companies, including Netflix, are not Agile across all of their company functions there are some modifications that must be made in those non-Agile functions [4]. Agile causes roles for employees to change so the company value of Agile and the adoption of Agile principals throughout a company matters [4]. This means functions like HR and finance will accept Agile processes whether or not their team members are integrated into Agile teams [4]. These functions need to have less levels and these teams often create a boundary inside of which Agile teams can do whatever they wish [4]. At its heart, Agile culture is about collaboration and empowerment for all employees.

2.1.4 The Pros and Cons of an Agile Project

Agile is profoundly different from hierarchal Waterfall management [11]. Not only is Agile hastening growth and improving managers but also is integrating people from different functions onto one team [11]. Agile is being utilized in all areas from National Public Radio to fighter jets to human resources [11]. However, it is important to keep in mind that although there are many advantages to Agile there are also some disadvantages to keep in mind.

2.1.4.1 The Advantages of an Agile Project

As established in Chapter 1, the United States Air Force acquisition process has a multitude of problems that have been inefficiently addressed by current processes. Not only are many projects over cost and schedule but they struggle to deliver to technical expectations [3], [6]. Fortunately, these problems seem to be addressed by utilizing Agile methodologies. For example, a study found that 50% of design is done outside of the design phase [13]. Although there is a wide variety of reports it is difficult if not impossible to find a study where Agile performs worse than Waterfall [3]. Most reports show that Agile is better on cost, schedule and/or quality [3], [5]. In some cases Agile has been found to increase team efficiency [2], [17]. Agile has been confirmed to increase client and coder contentment [2], [4]. One study analyzed a multitude of projects to better compare Waterfall to Agile methodologies [13]. This study found statistically significant evidence that the more Agile methods used were the more successful the project was, the happier the stakeholders were and the more efficient it was [13].
There also is documentation that Agile minimizes waste and creates the best value for customers [11]. It has even been shown to build respect between team members [11]. One review analyzed over 300 studies and found that on average Agile was 30% better at cost, 90% better at schedule, 97% better quality and 400% better satisfaction [14]. These are truly the headliners for the many advantages of Agile – but there are arguably many more important advantages below the surface.

Although Agile boasts many advantages beyond financial one of the most vital is managing change well [4]. As mentioned earlier, Agile encourages keeping the options open, especially for architecture, waiting until the most relevant information is available [3].

“That’s the agile in practice: big ambitions and step-by-step progress. It shows the way to proceed even when, as is so often the case, the future is murky [4].”

This process not only allows critical faults to be discovered early on, makes quick change possible, and makes integrating user feedback easy and fast [3]. This is vital since approximately 80% of Waterfall projects have erroneous requirements [5]. “Change tolerance” is also essential for understanding the customer’s perspective, since their perspective will change [5]. Agile teams first prioritize on the customer’s primary needs with a strong emphasis on customer collaboration [19]. The iterations make it possible to have frequent feedback to integrate in those changes [5]. This creates the Agile benefit of being able to respond to constantly shifting client needs [20]. Agile also engages in the customer in a more intuitive way. Rather than documentation Agile uses things like models to get better insight from customers [5]. The Agile Manifesto itself encourages empowered teams, effective communication, adaptation to change and frequent iterations [19]. This creates deeper understanding and cooperation between all shareholders [20].

2.1.4.2 The Disadvantages of an Agile Project

Although there are many advantages to implementing Agile – it is not necessarily the best fit for every project. Rico actually found that when comparing the best Agile methods to the best Waterfall methods that in terms of return on investment (ROI) they are equal [14]. Safety requirements often require more up-front planning, more reflective of Waterfall then Agile methods [13]. For example, in a Nuclear Power Plant using Agile methods could result in
irreparable damage by challenging failure [16]. Applications that rarely, if ever, have changes in code would not benefit from Agile [3]. In the DoD, the DSB found that areas such as low-level mission critical flight control systems, digital engine control systems and legacy systems near termination were unlikely to benefit from a switching to Agile from Waterfall [3]. In addition, upper management loses some their control mechanism with less upward transparency [5]. In these situations a balance between Waterfall and Agile may be needed since too much planning for a project can be just as disastrous as too little planning [13]. Additional types of projects that do not warrant Agile are more routine operations [11]. It is important to determine if the benefits of Agile outweigh the cost of the training and organizational change which are very expensive [11]. Since Agile focuses on empowering motivated people – teams must choose to adopt it [11]. Trying to force Agile will result in failure [11].

One of the biggest challenges with Agile is that it was designed for small teams – the larger the project the bigger the challenge [2]. Sizeable projects need more coordination, have greater interdependencies which often means more documentation – which is not very Agile-like [2]. Failure to adapt Agile processes well often leads to abandoning Agile practices [2]. This often is a loss of benefit; even non-Agile projects can gain from adapting some Agile aspects [2], [3]. Large scale Agile execution is largely undocumented in literature – making the challenge great [2]. The transformation from Waterfall to Agile was rarely if ever documented in the few reports that did focus on large-scale Agile [2].

2.1.5 Defining Large-Scale Agile

Despite having an unclear path forward, large enterprises are not willing to step away from the potential advantages listed above and continue to pursue Agile practices. There are a multitude of ways to define what large-scale Agile is with the number of people being the most common, however research has not yet come to a consensus [2]. Some other ways include the number of team members, the number of teams, budget size, the number of lines of code or project schedule [2], [3]. This does not add much clarity when 300 team members is defined as large and 50 team members is also identified as large in one report and small in another [2]. The DoD still uses number of lines of code even though this varies greatly between languages and is appearing to be exponential with time regardless of the project [3]. Despite the unclear definition of what exactly is large, there are similar challenges to adapting Agile to different large applications.
While many large businesses have succeeded in implementing Agile, many others have failed [4]. One of the keys to successful large scale implementation was found to be implementing a pilot program [2]. This is further collaborated by the fact that many enterprises found Agile more difficult to implement than expected [2]. Pilot programs helped persuade management and created company faith in Agile ways [2]. However, at other times Agile falls short of company hopes – this often means non-Agile company areas need to be adjusted [4]. During the transformation non-Agile business areas often delay Agile teams with bureaucratic processes [2]. Large-scale Agile implementation requires a careful balance of conformity and customization [2]. Conformity is needed for success however teams also need to be empowered to select the tools that add the most value for their project [2]. Research is ambiguous on timing and methods for eradicating old procedures [2]. However, it is important to find hold-ups to the Agile transformation and address the root cause [2]. Otherwise the Agile team does not find a realistic path forward, instead it has all obstacles magically removed [2]. Actual circumstances are important for companies to learn [2].

Another key in large-scale implementation is utilizing a team of teams [4]. With Saab’s Gripen fighter jet the team morning standups escalate system of system issues [4]. This has proven to be fruitful and resulted in better collaboration between designers and test pilots with colocation [4]. However companies who do large-scale Agile find that choosing and customizing an Agile approach was often seen as essential to success [2].

There are a variety of frameworks to choose from for Large-scale Agile [21]. For example, scaled Agile has been successfully implemented by Lockheed Martin even for multi-year programs [21]. This has been very successful for them and their clients [21]. They noted having individuals experienced in the chosen Agile framework is necessary for true success [21]. Another common issue for large-scale products is that within the system of systems, not all systems are using Agile and some systems are potentially being developed by a direct competitor [21]. Multiple-horizons is key with the system of systems where everything is not using Agile methods [21]. The more often the systems of systems are integrated the better, but sometimes the best that can be accommodated is annually [21]. Lockheed Martin has successfully adapted this framework to integrate their customer needs for critical security products as well [21]. In
fact Agile has been used on complex projects such as NASA’s Apollo and deep space probes [14].

2.1.6 Agile in Regulated and Safety Critical Environments

Agile methods can actually improve security [21]. Security critical product begins the same as any other Agile product development with a backlog of stories [21]. In this case security stories are part of the backlog [21]. There are also a series of steps that need to be taken in the PDCA Agile cycle such as threat modeling, utilizing defensive coding and secure design principles [21]. It is important to note that we still need high-level requirements and detailed design but the size of increments for the design, development and test are the same [21]. These items, much like Agile requirements are more “activities” that are likely to be redone and re-evaluated rather than done once as in Waterfall [21]. Security is included as a focus from the beginning and is part of every activity and usually a security specialist is included as a key team member [21]. Part of a secure design is finding areas such as those with a high number of interfaces making them more vulnerable [21]. Testing is critical, especially layered security test levels [21]. It is important to remember that there are risk based security tests, because testing all cases is impossible [21]. One example of testing is resiliency testing and using triage for the results [21]. Most government projects with critical security needs requires accreditation that needs to be done before deployment [21]. Often operations is a team member in secure deployments [21]. Secure deployments also often include secure infrastructure, only releasing one image in one container, or toggling a feature for a limited audience [21]. This enables getting back to a known state fast [21]. Operations result in feedback going into the product backlog [21]. When a defect is found there is complete traceability from requirements to the codebase and integration timelines [21]. This makes it simpler to handle defects found after a few releases [21]. These steps result in secure Agile systems [21].

“Embracing a culture of continuous improvement results in secure systems” [21].

This is how Lockheed Martin has successfully created secure, cyber security robust software for clients including the government [21]. The Intelligence community has even required Agile methods for development of their products [21]. This framework has enabled Lockheed Martin to utilize Agile on projects with hardware in parallel with software [21]. For example, it has
been utilized on the integrated fighter jet group such as the F-16 and F-22 [21]. The team will build every two-weeks and get feedback at that cadence [21]. Of course the software is deployed at a less frequent basis [21].

Below are two more specific examples of different DoD related efforts that have successfully utilized Agile in systems that require high security, high safety or both.

2.1.6.1 National Security Agency

Agile is utilized at the National Security Agency (NSA) [3]. To ensure security needs are met the NSA maintains a software factory that is a hybrid of business tools and internal security processes [3]. Utilizing this software employees can securely collaborate from diverse locations [3]. Since the NSA desires to own the intellectual property contractors are paid hourly with daily participation from government employees [3]. Metrics and procedures are also controlled by the NSA [3]. An undesired result is there is little investment by contractors since they do not own any intellectual property [3].

2.1.6.2 SpaceX

DevOps and Agile are utilized at SpaceX for safety critical systems including the USAF Space Launch [3]. Although DevOps is mostly used Agile is used for critical internal systems [3]. The government provided specifications for the launch mission [3]. SpaceX wrote specifications on how it would meet those launch mission requirements [3]. As long as the product is launch certified SpaceX can freely develop since the contract is a launch as a service [3]. Continuous deployment and testing are utilized by SpaceX [3].

2.2 Defining an Agile Team

Although many projects can work in Agile, including those in safety critical environments – it is important to pick the right projects to start [12]. Likewise the first teams should include the right people, such as those already exploring Agile [12]. A champion is also likely needed because change is always a challenge and cultural change is the most difficult [12]. The Agile teams and those around them greatly influence an Agile projects success or failure.
2.2.1 Agile Team Makeup

Many corporate teams are siloed by roles or functions, however Agile teams of all different roles collaborate together [22]. The traditional setup leads to diverging priorities resulting in conflict [2], [22]. Traditional groups are also concentrated on documentation and planning [19]. New Agile teams are most successful when they are focused on the principles of Agile rather than the procedures and when everyone is aligned to the same goal [2]. Leadership should remove obstacles for Agile teams rather than creating an Agile directive [4]. This enables a grass root effort which was seen as another key factor of success [2].

Within the Agile team itself, the team works to create a plan for a project [11]. They break down the highest priority tasks, determine how much work to start and how to divvy it up [11]. The team also clearly defines what being done looks like [11]. Every day the Agile team meets to quickly work through conflicts and obstacles [11]. Conflicts are handled through experimentation rather than verbal arguments [11]. After a sprint the team brainstorms ways to make the next sprint better and tackles the next item [11].

As discussed in Chapter 2.1 team members are empowered and this is reflected in the team makeup. Teams should be independent, this means volunteering, self-organizing, self-motivating, and self-governing [2], [4]. Team members may be guided by an initiative owner who guarantees everyone is involved and coaches members [4]. An initiative owner is also responsible to the business and customers for the product delivery as well [11]. Because of this they often manage stakeholders and prioritize based on value delivery [11]. An experienced team member should coach development teams on critical subsystems [17]. A process facilitator often helps coach the team through the break-down of tasks and time estimates [11]. Teams should be responsible for their work, which means few, if any approvals should be needed including funding [4]. Program managers may feel left out as they are no longer the regulator and planner [2], [19]. If they become a Scrum master they may struggle to understand their new function is facilitator not regulator [2], [19]. A master developer is required when designing a highly complex system to facilitate coordination across subsystem teams [17]. In addition to a design team, having an Agile team specifically for deployment and integration can be beneficial [12]. This would involve having subject matter experts in testing, operations, amongst others to help ensure that deployment goes smoothly[12]. An example of a deployment and integration
team composition is shown in Figure 2.5. Agile teams also need space for daily stand-ups and should be seated with their project teams [2]. When a company is well architected their teams will align well with the client experience and the business model [4]. This is vital since otherwise it can make dependencies across teams challenging especially if they have different velocities [2].

The developer teams are not the only organizational teams which are different in an Agile organization. In order to enable development teams to work directly with customers senior managers should be selling longer term strategies [4]. Unfortunately middle management roles are unclear in Agile which can lead to inappropriate controlling actions [2]. Leadership handles Agile better when they embrace the Agile values [4]. For example rather than creating a steering committee a working group is more Agile-like creating a backlog of opportunities that can be broken into pieces [4]. Having a champion for Agile is also beneficial to help remove barriers and take on risk [12]. This enables a new Agile team to succeed and makes a difference in their likelihood of success [12]. Although roles and functions are often different at all levels for Agile teams compared to Waterfall, all can benefit by focusing on the Agile principles as guidance.

2.2.2 Company Support for an Agile Team

As mentioned above, one of the keys to Agile transformation is having the right kind of management support. This means having the resources to guide, educate, and remove hurdles [2]. Management should have Agile training and openly support the process [2]. If employees are transitioning to Agile processes their workload needs to be adjusted to be lighter to prevent slippage into old ways [2]. When starting a new product leadership needs to be prepared to re-organize teams as the project progresses [4]. Team members should be carefully selected and
those with some Agile experience, different dispositions and capabilities should be taken into consideration [2].

Leadership is not the only key component in the company support; human relations (HR) company support for Agile is unique and also important. The company should become very team centric in their HR practices – emphasizing the team over self [4]. The culture for the employees should support failure, avoid condemning and instead see it as innovation [22]. Employees that lead a team well should be rewarded publicly, not with money but with something visible like an important project [4]. Additionally, employees should no longer show growth through job titles [4]. Employees should have mini feedback sessions throughout the year instead of one annual one [4]. By a company modifying HR, it ensures that employees are encouraged to truly embrace Agile values at all levels.

2.2.3 Agile Team Member Training

Training is also critical to enable employees to embrace Agile at all levels. Training is a vital component for new Agile team member, this is even more true when a company is transforming from Waterfall practices to Agile ones. Training in Agile is not only needed for developers, it is needed for all stakeholders, especially for contractors [2]. Training was found to be essential to keep employees from returning to old ways in large enterprises [2]. After preliminary instruction it is essential to have coaching to help teams navigate their way to success [2], [4]. Since hands-on experience is the best way to learn guidance is needed to help steer the course, correct mistakes and prevent harm [2]. Large organizations often do not have enough coaches to meet needs which explains why in some cases pilot programs do well but further incentives fail [2]. Although often overlooked, coaching seems to be just as vital, if not more so, then training for Agile implementation.

2.2.4 Customer and Agile Team Relationship

Just as training is vital for employees to successfully implement Agile, fostering a strong customer relationship is vital for the project to successfully implement Agile. The consumer should be directly connected to all Agile work [17]. The consumer should also have a direct feedback loop to the work being accomplished [17]. In contrast in Waterfall documentation and procedures are used to communicate user needs to the developer [17]. Because clients perform
miserably at describing solutions but are good at describing their problems Agile is much more effective [17]. To make matters worse a consumers circumstances are constantly changing meaning their problems are changing [17]. Rather than working through requirements documentation a developer has a prioritized list of features direct from the customer [17]. Iterations are used to adjust to user feedback [3], [17]. This process often results in user buy-in after a few iterations [3], [17]. Client request per iteration needs to be limited [17]. Following this Agile process results in developers making better, comprehensive decisions, using not only their technical understanding but also their better understanding of the clients’ needs [17].

2.3 Agile Product Requirements, Testing and Implementation

Waterfall style requirements ultimately result in low-level decisions being made before knowing if the high-level decisions are the right ones [17]. This feels safer because it decreases the complexity of the conundrum [17]. However omission and mistakes are pricey because of the cost of change [17]. Also, latent interdependencies within software design is where much of the risk is [3]. This is why simultaneous design is better [17].

2.3.1 How Requirements are Developed

It is important to note that Agile does not relinquish any planning at the beginning of a project [13]. There is important information that must be extracted from the customer [13]. As mentioned in Chapter 2.2 requirement documents are a poor way to communicate customer needs [17]. Instead things like prototypes and models are more intuitive if both the developer and user understands them without assistance [17]. Agile does not lack planning but rather disperses the planning throughout the duration of the project [13]. Almost as important as customer needs is the need for a system that can handle change [17]. The circumstances for a customer will change, it is just a matter of how quickly [17]. Agile’s emphasis on simultaneous design allows designs to be compared [17]. Simultaneous design is necessary since rarely, if ever, a design is right the first time [17]. Proving that there is planning in Agile, planning was found in one case to take nearly half of the client’s work contributions in Agile projects [13]. In some sense software requirements are never finished, over half of software project spending is after it is fielded [17]. Exacerbating this is the fact that user needs today do not often match strategic goals [17].
Dissimilar from Agile, high-level requirements are mostly absent [17]. For larger projects to succeed high-level requirements need to be broken into appropriate sized features for iterations [17]. Unlike in Waterfall where requirements map directly to cost, budget should not be tied to features [4]. This means that requirements and budgets do not work on an annual cycle that is used to encourage meeting objectives [4]. Instead finances is seen as an education opportunity much like venture entrepreneurs [4].

2.3.2 When Requirements are Developed
At the beginning of a project high-level requirements should be developed immediately followed by conceptual architecting [17]. Additionally all safety and security considerations should be deliberated [17]. Decisions that have an extremely high cost escalation ratio should be limited and seen as constraints [17]. Multiple concurrent architectures should be done [17]. These should be evaluated by experienced personnel to ensure there are no pertinent omissions [17]. Not only will concurrent designs lead to faster, better results but also will provide protection against making an erroneous decision [17].

“Concurrent design defers decisions as late as possible. This has four effects:
- Reduces the number of high-stake constraints
- Gives a breadth-first approach to high-stakes decisions, making it more likely that they will be made correctly
  - Defers the bulk of the decisions, significantly reducing the need for change
  - Dramatically decreases the cost escalation factor for most changes [17].”

To further mitigate risk from erroneous requirements, selections that are high-risk for change should be collected together if possible to mitigate change proliferation throughout the entire system [17]. Additionally, as soon as an mistake is found it should be fixed to minimize proliferation [17].

2.3.3 How are Products Tested
At a minimum automatic testing should be run weekly, ideally it should be run daily on the new pieces of code [17]. The entire system should be tested as frequently as possible [17]. On a large project testing the boundaries can be challenging due to the coordination and number of
individuals required [2]. Whenever possible small pieces of code should be released for testing [17]. This prevents a bottleneck where despite quick coding testing is far behind [17]. It is important to be able to test subsystems separately from each other [17]. This can be done by simulating other subsystems feedback [17].

In Agile there are different types of testing that are utilized. Agile is most known for its frequent customer tests that give developers feedback if they are meeting clients' needs [17]. Load or memory testing may be needed for a large project although that is less common in Agile [2]. Regression tests should also be ran regularly for business rule compliance [17]. Testing is crucial for system architecture and coding; quality tests will even find the source of inadvertent emergent qualities [17]. Testing ensures the emergent qualities are desired, communications function and makes it possible for last minute decisions [17]. This is why customer tests should not wait for a completed product [17].

In contrast, in Waterfall often a large volume of change will stress the testing and quality assurance personnel [12]. Automation is necessary to keep them from being the bottleneck [12]. Additionally, team metrics need to be aligned to the new goals [12]. A case study showed a program was able to save $1.5 million [12]. Although not always done in Agile, Agile DevOps and DevOps try to implement as soon as testing is finished.

2.3.4 When are Products Implemented

Whenever feasible the spanning system should be implemented as soon as it is complete [17]. Before subsystem design begins the interfaces for them should also go into production [17]. Unlike Waterfall implementation, Agile implements tested code in increments after an iteration [20]. In Waterfall product implementation was often a key signal of a transition of a program – whereas in Agile it happens frequently and there is not inherit finish line. Product implementation can be challenging – in one case deployments took 3 months and never went right the first time it was released [12]. This is likely a sign that Agile implementation can be improved. In this case, after transforming manual steps to automation in the implementation process and improving consistency with Agile DevOps the case was able to deploy products with 20 minutes of preparation instead of 3 months [12]. Quality was also increased making it possible to have consistently zero issues during implementation [12].
2.3.5 Agile vs. Waterfall Product Requirements, Testing and Implementation

Product implementation in Agile inherently changed the sub context of product implementation compared to Waterfall. This is equally true for requirements and testing along with cultural changes that come with a change in methodology. Below is a table summarizing these differences.

Table 2-1 Traditional Waterfall vs. Agile Source [20]

<table>
<thead>
<tr>
<th>Process</th>
<th>Waterfall Approach</th>
<th>Agile Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning and Scheduling</td>
<td>Pert/Gantt, detailed and upfront; fix scope, estimate time, and resources</td>
<td>Release and iteration plans updated throughout; fix date, estimate scope</td>
</tr>
<tr>
<td>Requirements and Design</td>
<td>Detailed and upfront</td>
<td>Continuous, emergent, last responsible moment</td>
</tr>
<tr>
<td>Implementation</td>
<td>Code in parallel, follow plan, change control, deliver at end of phase; test afterward</td>
<td>Code and test; deliver incremental working software each iteration</td>
</tr>
<tr>
<td>Test/QA</td>
<td>Detailed test plans; test after implementation phase</td>
<td>Continuous integration, build and test</td>
</tr>
<tr>
<td>Management Culture</td>
<td>Hierarchical and contractual, “command and control”</td>
<td>Servant leadership, collaboration, flat organization</td>
</tr>
<tr>
<td>Measures of Success</td>
<td>Conformance to plan or contract</td>
<td>Working software, satisfied customers and team</td>
</tr>
</tbody>
</table>

Because of how the Waterfall methods handled things like requirements these underlying assumptions have gotten wrapped into “higher” USAF guidance. For example, the USAF has Preliminary and Critical Design Reviews which expects detailed plans and documents which were inherit in Waterfall but are not in Agile [20].

This chapter has described what defines Agile and companies that embody it in addition to differentiating it from its predecessor, Waterfall. To better understand the USAF Agile transformation, next is an analysis of the current USAF acquisition process where a mix of Agile and Waterfall methodologies are being utilized.
3 Current Description of the USAF Acquisition Process

As mentioned in Chapter 1 the United States Air Force has started moving towards embracing Agile as a strategic fit for many acquisition projects. As with many transformations, the focus seems to be on the desired end results.

“Law Number II: If today were as good as tomorrow is supposed to be, it would probably be twice as good as yesterday was [1].”

This is too often true, and careful steps must be taken to avoid this pitfall. Despite the optimism about Agile implementation in the USAF there is uncertainty on the path forward. For this reason tools from the Nightingale and Rhodes’ book, Architecting the Future Enterprise, were used[8]. Their method implements a holistic approach for enterprise transformation [8].

In many ways the predicament of the USAF is reflected in the description of an enterprise attempting a transformation – endeavoring to transform because of compelling necessity, issue or opportunity [8]. What is humbling and concerning is the number of enterprises that floundered or failed while making enormous efforts to change [8]. The common cause of these failures is neglecting to think holistically about a complex system [8].
3.1 Motivation for Investigating the Enterprise Landscape

It is extremely challenging to comprehend an entire enterprise [8]. Nightingale and Rhodes found that a solution was to use multiple diverse lenses to assess an enterprise [8]. This reveals multiple perspectives of the enterprise which decreases the likelihood of tunnel vision for one aspect [8]. It also reduces complexity so the entire enterprise can be studied [8]. This results in the enterprise element model shown below [8].

![Enterprise Element Model](image)

Figure 3.1 Lenses for Different Views of an Enterprise Source [8]

Each of the elements are intertwined with one other [8]. That is, the sum of the elements is not same as the entire enterprise [8]. An analysis of the current landscape is just the beginning of the recommended process for completing a transformation [8]. However, the enterprise element model is a great tool for holistic analysis of the United States Air Force enterprise.
3.2 Ecosystem

Like biological ecosystems, and enterprise ecosystem’s components are all intertwined [8]. It is important to understand these links and how it could affect the enterprise’s ability to survive [8]. A part of the ecosystem analysis is finding significant uncertainties that effect the enterprise both now and in the future [8].

3.2.1 Definition

The ecosystem is the exterior scene where the enterprise lives [8]. Because of feelings of familiarity care must be taken to not make assumptions [8]. Extra precaution should be taken to understand value-driven factors, these are things that will alter attitudes about the products and services delivered [8]. It is also important to consider how these may change in the future [8].

For ecosystems Nightingale and Rhodes recommend considering seven different factors, finding that often at least one of these factors changing (or the threat of it) causes enterprise transformation [8].

“Ecosystem factors may include economic, political, environmental, resource, technology, and market-related factors [8].”

It is important to identify these factors, particularly the principally influencing ones to gain insight of the enterprise [8]. Finally it is important to keep in mind that just as the world is constantly changing, so are these ecosystem factors [8].

The below table summarizes the ecosystem factor analysis described below in the following sections. Political, regulations, resource, and technology factors were all classified as major factors due to current or potential future impact on the ecosystem. Economic, market-related, and environmental factors were determined to be, and likely continue to be, minor factors in the ecosystem.
### Table 3-1 Summary of Ecosystem Factor Analysis

<table>
<thead>
<tr>
<th>Ecosystem Factor</th>
<th>Classification</th>
<th>Summarizing Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Political</td>
<td>Major</td>
<td>US adversary aggression could quickly shift, having a large impact on the ecosystem. Congressmen are motivated to keep and create jobs in their state although this has minimal impact on mainly software programs [6].</td>
</tr>
<tr>
<td>Regulations</td>
<td>Major</td>
<td>Mixed debate on effects. Many push Agile but few truly embrace Agile methodologies.</td>
</tr>
<tr>
<td>Resource</td>
<td>Major</td>
<td>Contactor heavy workforce, especially in engineering [6]. Lack of systems engineers in the defense workforce [6]. Deficit of employees who want to work in defense [6].</td>
</tr>
<tr>
<td>Technology</td>
<td>Major</td>
<td>Integration of new technologies that have never been utilized [6]. 90% of the DoD’s weapon systems rely on software [5].</td>
</tr>
<tr>
<td>Market-Related</td>
<td>Minor</td>
<td>Defense contractor mergers are concerning to the DoD, with uncertainty of impacts [23].</td>
</tr>
<tr>
<td>Economic</td>
<td>Minor</td>
<td>Defense spending effects the US economy [24], [25].</td>
</tr>
<tr>
<td>Environmental</td>
<td>Minor</td>
<td>There are requirements to recognize and plan for limiting and safe discard of potentially hazardous materials from weapon systems [26]. Continue to monitor, potential to become major factor.</td>
</tr>
</tbody>
</table>

#### 3.2.2 United States Air Force Acquisition Ecosystem

As demonstrated in the above table, the USAF has a unique ecosystem that differentiates it from most commercial firms. However, like all ecosystems, its elements are intertwined. After analyzing each factor individually, the whole view becomes clearer after taking a step back. In the following sections the economic factors’ detailed analysis is discussed.

##### 3.2.2.1 Economic Factors

Although the health of the US economy certainly does effect government spending – most are focused on how defense spending effects the US economy then the other way around [24], [25]. How much defense spending influences the economy is hard to say, however many experts agree
that it is influential [24], [25]. However, Defense spending is highly linked to political factors [24], [25]. An interviewee also mentioned that the economy is often a source of innovation, creating a symbiotic like relationship. Thus, economic factors are classified as a minor factor for the USAF Acquisition Ecosystem.

3.2.2.2 Political Factors

Political pressures make it nearly impossible to cancel a program, only 5% of DoD Acquisition programs get canceled [6]. For a congressman program cancelation can lead to state job losses; which often causes a loss of votes and ultimately risks his/her reelection [6]. Programs that should be stopped or never funded live-on because lobbyists and defense contractors successfully persuaded congressmen to continue it [6]. However, one interviewee mentioned that this has limited influence on software only projects which do not generate as many jobs or as much funding as a hardware manufacturing line would. It also important to note the amount of money that these programs can generate. For example an acquisition category 1 (ACAT I) is estimated to spend over $2.79 billion in fiscal year (FY) 2014 constant dollars [27]. An ACAT II is estimated to be less than $2.79 billion in FY 2014 constant dollars, but exceed $40 million in FY 2014 constant dollars [27]. Political factors that influence USAF weapon development is not limited to US only politics and spending. For example, mounting aggression from North Korea, was expected to make budget legislation for defense easier to get approved [28]. Additionally, the US has been engaged in wartime-like efforts for over 25 years [29]. As one interviewee noted this could be considered more geopolitical than political factors.

Some of these political factors, such as US adversary aggression could quickly shift having a large impact on the ecosystem. However, congressmen being politically motivated to keep and create jobs in their state is unlikely to change and has a large impact on large programs. Regardless, political factors have a large impact on the ecosystem of USAF acquisitions.

3.2.2.3 Regulation

Here the term regulation is being used generally which is not necessarily reflective of the distinctions made in government. Laws, regulations and guidance all effect acquisitions, however they are written at different levels and have differing levels of influence. For example,
laws must be followed while guidance is highly encouraged to be followed but is not compulsory. In this case the category of regulation is being used to encompass all three, laws, regulations and guidance.

One of the main guidance documents for all acquisition programs is the DoDI 5000.02. It is just short of 200 pages, despite clear intentions in the latest update to make the process more tailorable and flexible to meet a more Agile methodology [27]. However, even in the example model there is heavy reliance on milestones and using them to limit what can be done. For example, Operational Test and Evaluation (OT&E) does not occur until after milestone C [27]. The model for a mainly software program is below in Figure 3.2.

Figure 3.2 Model of Commercial Software Adaption Source [27]
There is also a model for the possibility of incrementally deployed software in DoDI 5000.02 [27]. However, it is aimed at mainly adapting commercial software to defense needs rather than a truly new software. But it is a good step in the right direction. This model is shown below in Figure 3.3.

Figure 3.3 Model of a Mainly Software Defense Program Source [27]
There are also a multitude of other example models for different mixes of software and hardware programs with different parameters [27]. One of the potential concerns is the multitude of milestone B’s or C’s come associated with having multiple production and deployments [27].

Outside of the models, despite the ability to utilize more Agile methods, the metrics in the DoDI 5000.02 are also highly focused on metrics from Waterfall methods. Programs are held to schedules and costs [27]. If they go six months past scheduled dates or ten percent over value the program needs to be reapproved [27]. Additionally, tools which are ideal for Waterfall, not Agile, such as Earned Value Management are mandated at time intervals such as monthly, for contracts over 20 million dollars [27]. Another example is program management responsibilities are to make cost, schedule and performance as estimated at the baseline [27]. Programs have to know the entire expense for the total number of units they wish to purchase before it is established if it functions [6]. In contrast in the system engineering section for software describes an Agile like process [27]. The DoDI 5000.02 not only mandates adherence to the Clinger-Cohen Act but also stalls programs from moving forward if compliance is not met [27].

Per the Government Accountability Office, following large numbers of regulations that dictate what you should and should not do gives some personnel in the DoD the belief that risk is controlled [6]. An interviewee thought that regulations are not inhibitive; individuals tend to lack a true understanding of the regulations. However, the interviewee did state that cyber security regulations can be challenging. Another interviewee felt that as programs continue to transition to Agile methodologies the areas in which regulations fall short of Agile needs will be identified. This interviewee felt that many regulations are using Waterfall but saying Agile terms.

There are some regulations which give some ability to control the process better [30]. There is special funding for urgent needs [20]. However, most programs would have to use another route. There is an Accelerated Acquisition Program model for when US adversaries are surprised with technology [20]. Or if it needs to be in the field even quicker there is the rapid Acquisition Activity which enables fielding within two years [20].

Most of the regulations which control USAF acquisitions are legislative or created the Department of Defense level. These requirements are the initial architecture of the United States
Air Force acquisition processes. As such regulation factors were classified as a major factor in the ecosystem.

3.2.2.4 Environmental Factors

Environmental Factors effect defense projects with requirements to recognize and plan for limiting and safe discard of hazardous materials [26]. In software projects with no hardware this due diligence is not required [26]. As an external factor, environmental concerns are not currently a major factor effecting USAF acquisitions.

3.2.2.5 Resource Factors

Although not often discussed in most acquisition papers, the USAF acquisition workforce is an interesting predicament. In the 1990’s there was a move in the DoD to outsource many of its science, technology, engineering and math professionals in an effort to save money [6]. It has escalated to the point that some offices have a one to one ratio of government employees to contractors [6]. This makes it difficult to oversee contractors with little internal expertise [6]. Perhaps one of the causes is that it is hard to open a new civilian billet according to one of the interviewees.

Additionally, there are concerns about young professionals lacking interest in defense work, making it challenging for both the DoD and contractors to find new employees [6]. One reason for this according to an interviewee is that many non-defense software programmers do not want to have a government clearance. Adding to the problem is a lack of systems engineers [6]. This was even cited as a main cause for unacceptable cost overruns that led to the program cancelation of a 2007 US Navy ship [6]. One interviewee mentioned that rather than industry competitive pay, pay is capped by time in service. Another interviewee mentioned that because of the lack of people who trained Agile government representatives there was a loss of control over the prime contractors doing development. There also is a lack of program managers with expertise with software and lack a proven track record [31].

Having a contactor heavy workforce, especially in engineering makes contemplating a transformation to an Agile organization interesting. An additional lack of systems engineers in the defense workforce working on complex systems could also make the transition more
complicated. These resource factors are especially important to consider since it takes considerable time to gain desired expertise [6].

3.2.2.6 Technology Factors

Much like most systems in homes today, approximately 90% of the DoD’s weapon systems rely on software [5]. The amount of software is increasing in weapon systems [3]. This has resulted in cyber warfare being here and the DoD is a target [5]. For example, there are multiple claims of stolen US weapon plans including the F-35 [5]. This is concerning, especially considering US adversaries know it can take up to ten years to deliver software [5]. Technology cycle times need to decrease in defense acquisitions [30]. To make matters worse US adversaries are obtaining capabilities quicker than ever [3]. Exacerbating things further is the integration of new technologies that have never been utilized [6]. Agile methods are seen as way to better respond to these threats than our current capabilities allow [3].

The growing amount of software in weapons systems is seen as one of the major motivators for the transformation. As shown in Chapter 2, Waterfall methods are not ideal for most software developments compared to Agile. Technology factors were identified as one of the major factors for the ecosystem.

3.2.2.7 Market-Related Factors

There have been multiple merger and acquisitions in the defense contractor community in recent years [23], [28]. It began with smaller defense companies, not the prime contractors, who are the big players in the defense industry [23]. Senior military officials would like to keep mergers, especially of primes, to a minimum, stating if possible they would like to restrict those types of deals [23]. Northrop Grumman, Orbital ATK, ManTech International and Booz Allen Hamilton are just some of the companies involved in these types of deals [23], [28].

With the noted sky-rocketing costs of weapon system development it is understandable why these mergers could be concerning to the DoD [23]. Consolidation of firms could also have positive effects such as more funds for R&D from shared over-head for the two firms. However, mergers could also limit options that may be needed or desired. Because of the limited impact it
would likely have on the weapon system development market-related factors were considered a
minor factor for the ecosystem.

3.3 Stakeholders

After analyzing the ecosystem, the stakeholders are another major element to be examined in an
enterprise transformation. The value that is provided by an enterprise is largely characterized by
the stakeholder’s views and the value exchange that takes place [8]. These stakeholders include
those within the enterprise, those who interact (even incidentally) with it and those who are
affected by the success or failure of the value creation development [8].

3.3.1 Definition

Many times, stakeholder analysis is focused on those stakeholders who are invested in the
enterprise [7]. These stakeholders are identified in Crawley et al.’s book Systems Thinking as
beneficial stakeholders [8]. Beneficial stakeholders both give and receive value [8]. There are
also charitable beneficiaries who get value without giving anything and problem stakeholders
who do not get anything but the enterprise needs something from them [8]. The different
stakeholder exchanges can be represented in a stakeholder map [8]. This map translates the
stakeholders to a system [8]. This system can be a useful model to better understand the
stakeholder network and the value exchange feedback loops that exist [8].

3.3.2 USAF Acquisition Stakeholders

A combination of personnel experience and literature review was used to compile a list of
stakeholders and their value flows. There are a couple of interesting findings from this mapping.
An interesting stakeholder is Congressmen, who do not get any direct value from the acquisition
programs; instead they only receive benefit through their local communities getting jobs.
Another interesting relationship are the non-program employees, such as testing, contracting,
finance that do not receive any benefit from the program but are vital to its success. One of the
most significant modifications due to interviews was the addition of the warfighter lead
command. Some interviewees had great success, while others found it impossible to get lead
command to understand the value of the true end-user in the acquisition process. In an article on
Agile in the DoD, it is noted that there needs to be greater access to end users and that they will
need to be more involved in the product development [20]. This is in contrast to the typical warfighter representative in traditional acquisitions, who has spent the majority of their time at the Pentagon [30]. Unfortunately, even in the guidance manuals this is not mentioned [30]. This map helps highlight how the value exchange results in some complex value networks. It is important to note that this model reflects how the architecture currently is; as things change these value exchanges may become different.
Figure 3.4 Stakeholder Map for Acquisition Program
3.4 Current Architecture

After analyzing the stakeholders, the current architecture is examined through view elements. The elements of the architecture will affect the transformation. Some architecture elements may need to be changed in order for a successful transformation while others can be left untouched. However, analysis of all the elements needs to be done to fully comprehend the current architecture.

3.4.1 Strategy

Starting in 2002 there has been growing recognition within the USAF that it could benefit from Agile like processes [32]. In 2014 the USAF published a 30-year strategy document which focuses on agility as the key to success in a future that is demanding more and more speed [33].

“We must purse a strategically agile force to unlock the innovative potential resident in our Airmen and turn a possible vulnerability into an enduring advantage.”

- General Mark A. Welsh III, Chief of Staff [33]

In a section focused on capability development it emphasizes Agile and mentions methods like prototyping and automation of procedures for productivity [33]. It also mentions Agile as a solution to interdependency problems so that multiple contractors can work on development [33]. It emphasizes the role of systems integration belonging to the USAF [33]. Finally it promises to update requirements processes and the acquisition system to make system modification or cancelation possible [33]. The strategy document also emphasizes a need to foster innovation even though it comes with risk, a very Agile concept [33]. However, the suggested implementation is somewhat non-Agile with a focus on schedule, and cost [33]. Fortunately it also encourages rewarding failure and risk-taking [33]. Also included in this strategy document are relationships the USAF wants to bolster [33]. These include Congress, universities and commercial companies, some of the noted stakeholders for our stakeholder map [33].

Following this in 2015 the USAF published its Strategic Master Plan for the following twenty years which included two imperatives, one of which is Agility [34]. Although Agility is not used
to describe solely acquisition or engineering processes the master plan does specify detail about capability development [34]. The capability development includes 6 main points which are:

(1) “Pursue modular, adaptable, and upgradeable solutions [34].” 
(2) “The Air Force will act as integrator [34]…”
(3) “Empower the Air Force as a customer [34].”
(4) “Incentivize innovative, competitive solutions [34].”
(5) “Inject Pivot Points to assist acquisition agility [34].”
(6) “Use experimentation for agile capability development [34].”

These six points are ways in which the USAF is planning on meeting its strategy imperative of agility. Modular design is intended to enable numerous developers for one solution [34]. This pairs closely with the second point – several developers requires a single integrator [34]. Increasing USAF ownership requires a technically savvy workforce which is emphasized by point three [34]. The fourth point switches focus, emphasizing how the USAF will continue to internally and externally invest in research and development efforts [34]. In order to best integrate these technologies and empower programs in decision making “pivot points” are created to enable vector changes by program offices [34]. Finally, it concludes with stressing the value of experimentation for inventive advancements [34].

3.4.2 Infrastructure

The USAF has fourteen locations with acquisition facilities, not including where headquarters personnel sit, which is in Washington, DC[35]. The following locations are across the United States: Hanscom Air Force Base (AFB) (Massachusetts), Wright-Patterson AFB (Ohio), Robins AFB (Georgia), Maxwell-Gunter AFB (Alabama), Eglin AFB (Florida), Offut AFB (Nebraska), Tinker AFB (Oklahoma), Joint Base San Antonio (Texas), Peterson AFB (Colorado), Kirtland AFB (New Mexico) and Edwards AFB (California) [35].

Some bases house certain functions – for example, Edwards AFB houses the USAF Flight Test Center [35]. Other bases, such as Wright-Patterson host a large variety of functions [35].
3.4.3 Products/Services

According to United States Air Force Acquisition website their mission is to:

“Cost effectively modernize to deliver capability to the warfighter when needed.” [36]

This has resulted in currently over 465 AF acquisition programs between research, development, sustainment and acquisition [36]. These capabilities are needed to support US defense and joint operations under a variety of often challenging conditions [34]. The aim of these capabilities is “Global Vigilance – Global Reach – Global Power” [33]. This means many of the capabilities are focused on air, space and cyberspace [33]. On the opposite end some of these capabilities will be utilized in areas with limited infrastructure, no laws and unlawful peoples flourish [33]. There is also a race to be at the front of ground breaking technology – dominance in this realm can be brief [5], [33]. This means that capabilities are often delivered in the form of a weapon system.

3.4.4 Processes

Many feel that the Waterfall process described in Chapter 2.1 is currently reflective of USAF processes have been recommended for the DoD starting about twenty years ago [5]. However directives and guidelines have not always suggested Agile processes [5]. The several milestone process was started in 2003 and is still in effect today [5]. Also still in effect today is the DoDI 5000 guidance [5]. Some believed that these documents fell short of Agile reform needs [5]. Some contractors feel that Agile processes cannot be adopted because of the documentation, progress reviews and incentives utilized [3]. For example, the DoDI 5000.02 does not officially list engagement with warfighters until after Milestone C, but states the program managers should have a strong sense of warfighter needs [30]. The president mandated an Agile acquisition process after this in 2009 and the USAF secretary also wrote on the need to use lean methods [5]. Unfortunately there is limited confirmation that Agile is being executed across the whole AF, although this could be due to secrecy [5]. This is unlikely since the DSB report found that some of the cases that claimed to be utilizing Agile were not in actuality [3]. It appears to be the exception when Agile is implemented [5].
There is some evidence of some Agile implementation in the USAF [3]. An example is the KC-46A tanker which uses a mix of Agile and Waterfall procedures [3]. Boeing segmented software into iterations for development [3]. This particular example utilized a fixed-price incentive-fee contract which means that if there is a surplus on fulfillment of specifications set in 2011, Boeing is held liable [3]. This creative solution is more of an exception than the rule with the recent DSB report finding that due to contract processes and defense culture execution contractors cannot have iterative capabilities [3]. Another example of processes being utilized to make Agile processes more possible is the focus on open systems architectures [29]. Currently most large weapon systems use reimbursable contracts with incentives for initiation of contracts followed by fixed-price with incentives for production and service contracts in sustainment [3].

Appendix A shows the pre-tailored process for acquisition programs. After receiving funding, on average it takes 110 months for a large military program to complete [6]. Often following the process is a way to avoid fault for failure of a program [6].

3.4.5 Organization

One of the most influential individuals for acquisition programs is the Program Executive Officer, PEO [36]. A PEO usually is accountable for one large program or a several related small programs. [36]. For smaller programs that are ACAT II and ACAT III the PEO is usually not the Milestone Decision Authority, which means he or she can decide if a program moves on to the next step, is canceled or needs further work on the current milestone [36]. Only a few years ago all decisions needed to be made at the PEO level [29]. Now however, nearly 70% of small program decisions are made below the PEO [29].

There is not a “single” chain of command for the United States Air Force acquisitions. The below image highlights what the high-levels of the chain of command looks like.
In addition to PEOs, there are capability and functional directorates [36]. Capability directorates include Information Dominance, Global Reach, Space Programs and Special Programs [36]. Functional directorates are Scientific Advisory Board, Contracting, Logistics and Product Support, Science, Technology and Engineering, Acquisition Career Management and Acquisition Integration [36]. Each of these directorates are run either by a civilian or military officer [36].

3.4.6 Knowledge

One of the primary ways the acquisition workforce is trained is through the Defense Acquisition University (DAU) [37]. DAU teaches essential acquisition information and can be a resource for on the job assistance [37]. Because of the Defense Acquisition Workforce Improvement Act, all members of acquisitions must be trained and certified in their career field at either Level I, Level II or Level III [37]. The DAU also supports continuous learning through their learning center which includes refresher training, hot topics and learning modules from the Harvard Business School [37]. Continuous learning is required by DoDI 5000.66 [38]. Every two years all acquisition personnel must earn eighty continuous learning points [38]. However, employees are encouraged by their supervisors to tackle this on a yearly basis [38]. These requirements are intended to ensure that the acquisition workforce is qualified and capable to complete their role in providing capabilities to the warfighter.
3.5 **Summarizing the Enterprise Landscape**

Since the USAF is such a large enterprise, a wide-breadth is covered when analyzing each of the elements that make up the USAF Enterprise landscape. By bringing together the analysis of different elements the whole begins to become clearer. For example, one of the ecosystem factors identified to affect the USAF Agile transformation was resource. This was due to a shortage of systems engineers in the defense workforce, in addition to other resource factors. The knowledge portion highlighted the focus on education of acquisition employees through DAU. After examining the individual elements, the whole enterprise can become clearer. By viewing elements that are pertinent to the USAF Agile transformation, the current status can be better understood. Now that the current status of the USAF transformation is better understood, an analysis of the current challenges being experienced during the transformation follows.
4 United States Air Force Transformation to Agile

“LAW NUMBER XXXVII

Ninety percent of the time things will turn out worse than you expect. The other 10 percent of the time you had no right to expect so much.” [1]

As Augustine’s Law points out rarely do things go as hoped. This was reinforced in Chapter 2, which mentioned that the transformation of any organization is challenging to pull off successfully. Despite many assuming that concept selection and future architecture is easy and the straightforward part of a transformation, it actually requires great labor [8]. In part this is related to the complexity of sociotechnical systems [8]. In fact large organizations were found to have a range of challenges with implementing Agile [2]. However, we cannot use this as an excuse not to transform. Transformation is necessary as an organization acclimates to a constantly evolving world [8]. In fact, the DSB report on acquisition methods found that traditional methods of acquiring software takes too much time and is too difficult to meet the needs of defense systems [31]. This is exacerbated by the short lifespan of commercial software [31]. With this in mind and an Agile AF transformation already underway, Chapter 2 on Agile and Chapter 3 on the current architecture of the USAF are married together to reveal the conditions for accomplishing an Agile transformation and what can be done to increase odds of success.

4.1 Challenges with Agile Transformation

Recently more literature is beginning to answer the call for more information on Agile methods, but rarely do these touch on the challenges instead focusing on successes [39]. There also is a lack of summarizing literature on a subject that is very pertinent to industry today [2]. Gregory et al. found from collecting challenges from Agile practitioners at conferences themes were “claims and limitations”, “organization”, “culture”, “teams”, “sustainability”, “scaling” and “value” [39]. Many of these topics are mentioned below in addition to some unique USAF related findings.
4.1.1 Laws, Regulations and Guidance

There is a mix of feelings on how much of a challenge laws, regulations and guidance create for Agile implementation, but it certainly is addressed in a number of documents discussing the DoD and Agile. Interestingly a report from 1962 on acquisitions found that Congress should not legislate to low levels [32]. A later paper found that programs are lengthened when Congress or the Pentagon micromanages causing other issues as well [32]. Laws could be improved to truly embrace Agile, however studies have shown that they are not Agile prohibitive [5]. For example the NDAA of 2010 section 804 requires Agile for IT procurement [5]. However, the paper stresses the value of “well-scoped and well-defined requirements” and contains a series of steps [5]. Additionally, there is the perception that the 5000 series conflicts with Agile but this is not true [5].

In 2007, Congress grew oversight to “unprecedented levels” including milestones, schedules, life-cycle cost estimates and prompts that mandate a congressional report [5]. This conflicts with other mandates that they made at the same time requiring Agile implementation [5]. The colors of money also make Agile development challenging as reported by multiple interviewees. These interviewees noted how there is either 3400 or 3600 money which is to be used for development or sustainment. The problem they experienced with Agile is that you are fielding a minimally viable product and then adding to it. So, interviewees experienced challenges with how to differentiate between development and sustainment in Agile. Differentiating determines if 3400 or 3600 money should be used. This is important because using money that was obligated for development but is actually sustainment is illegal. Continuous development does not trace well to funding and colors of money [3]. Despite this, small programs are able to work through these issues but this is not possible in larger programs [3]. This was noted as an issue by several of the interviewees.

Another interviewee mentioned that the Clinger-Cohen Act makes Agile implementation difficult. Clinger-Cohen was intended to address large problems with IT procurement in 1994 [40]. Clinger-Cohen was designed for Waterfall, focusing on meeting budget, schedule and scope [40]. This reflects the interviewee’s experience. There is some encouragement in the 2015 Federal Information Technology Acquisition Act, moving away from budget, schedule and scope [40]. However, the most current DoDI 5000.02 (updated in 2017) still states that programs
must comply with the Clinger-Cohen Act, essentially mitigating any difference the new act could of made [27].

### 4.1.2 Contracting

Like the issue above, contracting has a debate about how much of an issue it is – however it is clear that is affecting the process. Current contracting methods assume complete, detailed specifications for an entire system [3]. Contractors often lament that Agile claims do not match the documentation requirements [5]. This is often due to the milestone reviews that mandate unneeded paperwork [5]. Additionally, contracting methods do not couple well with operational necessity, risk opposition, or the crippled system that works at a slow pace and is very cost conscious so excellent employees are lacking [41]. Traditional contractors are therefore unmotivated to adopt agile [3]. With a few exceptions most DoD contractors utilize Waterfall processes [3].

Many policies for Agile implementation impede traditional contactors since they are not seen as innovative [41]. However, one interviewee commented that contracting as is discourages small businesses from pursing government work. This is because one program being canceled could ruin a small firm. Prime contractors feel that policies favor non-traditional defense companies discount the vast contributions these contractors made in Agile and does not recognize Agile scaling challenges [41]. However, DoD prime contractors have recognized that they are behind the non-defense industry. [3]. They are currently at different levels of understanding with Agile processes [3]. Contractors struggle to understand if the government wants innovation and if their business model will work in the shifting environment [41]. In contrast, newer companies are focused on a sole area and are uninterested in providing other services [41]. These companies try to empower their customer and lower the front product price [41]. This is in sharp contracts to most prime contractors who have horizontal and vertical integration [41]. Others defense contractors do not think Agile is a good fit for defense applications and are not interested in changing their methods [3]. Still others seem interested in adopting portions to help with cost and schedule [3]. One interviewee noted that traditional defense companies are upset by utilization of level of effort contracts with no prime contractor. This interviewee stated that this means no sustainment money for them, causing the upset.
Regardless, the DSB report found that contracting worked outside of the regular system to implement Agile processes and concluded that this was unacceptable [3]. However, one of the interviewees said that contracting had not been much of a challenge for them. Another, however, noted that contracting was a huge impediment to Agile implementation. Alternative acquisition methods, such as other Transaction Authorities, labs and special units are utilized to adopt new technology, instead of within traditional processes [41]. Instead of requiring the detailed requirements, Agile processes require a negotiable scope which allows the project to converge [17]. Additionally, Dare found that defense contractors needed prompt feedback from government representatives [32]. When they were unable to attain this the prime contractor made the decision to move forward [32]. This is a huge issue considering that the greatest impediment to Agile is the lack of trust between companies [17].

### 4.1.3 Learning Curve

Despite increasing research on Agile implementation there is still anxiety that research may not be applicable to real enterprises because of differences between research and actual companies [39]. This is concerning since nearly half of large scale enterprises in Dikert et al.’s literature review found that Agile is difficult to realize [2].

Challenges such as misinterpretation of Agile, scarcity of literature guidance, shoddy tailoring, regression to old habits and over eagerness make Agile implementation difficult [2]. Defense program offices tend to treat software acquisitions the same as hardware because of cultural standards [5]. They struggle to pinpoint key differences between software and hardware acquisitions [5]. The current DoD approach is capabilities-based rather than the desired threat-based acquiring model [3]. Users were unclear on their role in lean defense projects, unsure of the ramifications of requirements they wrote [32]. This was exacerbated since the lean analysis also found that acquisition members did not fully comprehend operational ramifications either [32]. Dikert et al. found that even experienced teams that succeed in training may struggle to actually realize Agile [2]. This was caused by a lack of a deeper comprehension, such as the purpose or the values [2]. Some examples of understanding challenges included showing partially complete work to clients, seeing Agile as a development that does not need design and perceiving it as fewer job opportunities resulting in competitions amongst team members [2]. Somewhat in contrast challenges from an XP conference included multi-team cooperation, how
to plan, communication with customers, along with specific large-scale Agile implementation challenges [39].

There is also a learning curve with leadership. When leadership was questioned about Agile in companies trying to implement it they knew basic Agile terms such as sprints [11]. However, they had not gone through training and lacked fundamental knowledge of Agile methodology [11]. Without realizing it these executives then continued to manage in ways that were not aligned with Agile making the efforts of Agile teams that worked for them more challenging [11]. For example, they would launch projects with strict deadlines rather than prioritizing projects [11]. Or they would schedule too many meetings preventing work from being completed [11]. Finally, they also would regularly override team decisions and add bureaucratic processes to prevent mistakes from being repeated [11]. Many of these are in a sense remnants from Waterfall, however, it was also included here since in this case Harvard Business Review was able to find that none of these leaders had gotten official training and instead relied on picking it up along the way [11].

### 4.1.4 Waterfall Remnants Still Permeates

Large Agile implementations struggle with how to break up high-level requirements into manageable sized stories [2]. Ambiguity in requirements in Agile is an issue especially for a large enterprises [2]. Gaps can result between short and longer-term planning with Agile’s focus on the short term [2]. Requirements ambiguity in large enterprises caused quality assurance teams to struggle [2].

This helps explain why defining requirements in advance is heavily stressed in the DoD despite multiple statutes recommending Agile for acquisitions [5]. Unsurprisingly this means specifications are not current, the warfighter would like them adjusted or are no longer needed [5]. For example, The USAF Acquisition Improvement plan at a glance appears to be emphasizing Agile implementation [5]. At a closer look it seems more like Waterfall then Agile [5]. (This could be due to it not being a software only guidance) [5]. These issues are reflective of problems seen in large-scale Agile implementation case studies. There were reports that leadership was too entrenched in Waterfall method [2]. Not only did this result in making adoption challenging it created a bottleneck for all projects [2].
With an exorbitant amount of documentation and authorization gates that are mandated Waterfall is comparable to bureaucracy [2]. In the USAF, programs depend on milestone reviews, heavy documentation, reporting and limited metrics to audit and determine contractor momentum and progress with the new technology [20]. When Agile is inadequately completed the documentation and gates demand two procedures [2]. Research on an organization that had begun an Agile transformation indicated that leaders favored advance detailed requirements [39]. Leaders tend to desire lots of documentation despite the team preferring relaxed Agile communication methods [39]. Leadership felt a loss of control over team and counted on formalized reporting structure in a Waterfall fashion [39]. One finding was that these challenges were exacerbated by rotating leadership [39]. Organization leaders needed to be educated on limited documentation and its benefits but the rotation increased demand for this training [39]. Hesitancy to adopt Agile is also present in the DoD where many senior leaders started in defense about the time Waterfall was mandated, seeing it as the default [5].

Research on an organization that had begun an Agile transformation also indicated that leaders favored meeting schedule, cost and scope [39]. This led to conflict when the Agile team focused on and delivered customer value, upholding Agile principles [39]. Some large organizations struggle with Agile implementation, keeping old commitments to deadlines causing Agile implementation to fail [2]. One interviewee mentioned that being forced to adhere to deadlines by leadership was a huge challenge for their USAF Agile program. Another example of comprehension issues were in continuing of Waterfall – creating mini-Waterfall iterations or managers overcommitting regardless of team velocity or reporting estimates unrelated to team velocity [2].

Waterfall appears to give tight control with barely any changes, although in it is actually highly risky [5]. In contrast Agile embraces changes which is perceived as exceptionally risky to program managers [5]. Leadership’s feelings of lack of control are further exacerbated when the Agile team allows the project to evolve rather than adhering to the predetermined scope and is viewed as a failure by management [39]. However, one interviewee commented that high-level USAF leadership is embracing Agile. Studies show that once the procedures have begun on a project it rarely is flexible [5]. Agile contrasts with the military, including acquisitions, which is hierarchical [5]. Agile instead emphasizes how everyone is equal and decisions are made at
lower individual levels rather than the highest ranking individual[5]. When a rewarding model is not updated and therefore is not team centric but instead individualistic this can cause failures from the team to truly embrace Agile [2]. These many Waterfall remnants can create change resistance at different levels within an enterprise.

4.1.5 Change Resistance

Over one third of industry organizations experienced change resistance [2]. This problem is seen not only in industry but also the DoD [5]. Within the DoD the main challenge with Agile is with the culture [5]. This is shown with the “...preference for familiar hierarchical control, risk aversion, and ignorance of the legal authority permitting agile methods” [5]. Workers dislike it when they do not understand Agile and they find it is difficult to change [2]. For example, one interviewee noted that when jobs change with the Agile methodology a person may no longer be the right fit for their job. When defense personnel struggle to pinpoint differences between software and hardware acquisition processes they resist change procedures [5]. One interviewee also noted that their team had lots of turn over from personnel transitioning, both civilians and military. This interviewee noted that despite a great, in-depth training that the whole office received that within a few years there were few people still there with the training. This made it difficult since none of the “new” people got the same training as the originals, although it was mentioned as a should-do item.

4.1.6 Non-Agile Functions

In Dikert et al.’s literature review they found that the third largest challenge for large scale enterprises was harmonizing non-development functions [2]. Some challenges in a research case study were related to the challenges of having one Agile organization within a company with a non-Agile company culture [39]. For example, Agile teams likely have to interface with teams that are not Agile or an oversight committee who may prevent Agile actions [2]. Changes in modeling, iterations and emphasis on features meet opposition when the standard is traditional life-cycle models [2]. One survey in literature found that the challenges with Agile implementation were not project based and more transformation of the enterprise based [39]. A holistic perspective was need to solve Agile related issues since no one challenge was contained to one area [39].
Through the interviews multiple non-Agile functions, were called out. Depending on the individuals’ experience determined how much of a challenge working with people from these non-Agile Functions were. These non-Agile functions included: high-level leadership, contracting (separate challenge category), finance, testing (separate challenge category), cybersecurity, human relations and prime defense contractor’s non-Agile functions. Many of these overlap with other challenge categories, but are worth noting here. One area that is not specifically covered in another challenge category is cybersecurity. Interviewees particularly noted that gaining an Authority to Operate, ATO, caused significant heartache. An ATO is an approval process that helps ensure cybersecurity, for example ensuring a system has appropriate controls, is handled by a separate chain of command than the program office [42].

4.1.7 Large-Scale Agile

Agile was originally intended for small teams and projects [2]. Introducing large-scale means increased coordination [2]. Additionally, there is generally more separation between the developers and the customers [2]. One of the interviewees mentioned that bureaucracy often gets in the way of the user feedback loop. In that interviewee’s particular experience, they found that it was challenging to get to users. Not only is there greater resistance to change processes in a larger company there also is a need for things like increased communication [2]. Additionally, large scale Agile challenges tend to be intertwined with others, such as non-Agile functions and change resistance [2].

4.1.8 Testing

Many interviewees mentioned that testing was a large challenge. Interviewees pointed out that within Agile there is no separate developmental and operational testing, however there is within the Air Force testing community. Most defense programs utilize testing right before release, a very different approach to testing than Agile where this is a continuous activity [20]. The interviewees also noted that this requires significant job changes for testing individuals. Interviewees were not sure how to handle USAF testing standards within Agile, but did feel that some type of checks and balances were needed, especially when Agile development is not done internally. One interviewee mentioned that their attempt to work with testing to integrate their processes has been met with resistance.
Challenges with testing are also seen in industry. For example, in one case a company had an Agile pilot team creating an application [11]. At this company the software had to wait in a long list to go through a computer code that tested for function and standardization [11]. After this then the application could be integrated to be implemented but took six to nine months [11]. This meant that despite the work the team did actual development time did not change very much [11].

4.1.9 Summary of Challenges

There are a variety of challenges documented in literature that are related to both Agile transformation and specifically to the USAF.

“For most programs, using Agile is approaching new territory, full of unfamiliar processes, lacking clear alignment to existing expectations, and/or one in which program stakeholders are unprepared to adapt to their changing roles.” [20]

The following table summarizes these findings.
Table 4-1 Summary of Challenges

<table>
<thead>
<tr>
<th>Challenge Category</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Law, Regulations and</td>
<td>Some regulations and guidance need to be accept Agile methodology, however this does not completely hinder Agile progress [5]. However, many Agile recommendations need to be rewritten or eliminated.</td>
</tr>
<tr>
<td>Guidance</td>
<td></td>
</tr>
<tr>
<td>Contracting</td>
<td>Contracting needs updated to make Agile easier to implement [3]. Defense contractors are unsure how to adapt to the Agile transformation [41]. New software firms are interested in government work but engage in very different ways.</td>
</tr>
<tr>
<td>Learning Curve</td>
<td>Even “expert” teams can struggle to implement Agile at first [2]. Defense employees are equally vulnerable, struggling to understand how to adjust for software versus hardware projects [5]. This is equally true for leadership [11].</td>
</tr>
<tr>
<td>Waterfall Remnants</td>
<td>Leadership feels loss of control without detailed specifications and loss schedule, budget and scope in the traditional sense [39]. Processes are still very Waterfall both those that are voluntarily utilized and those that are mandated.</td>
</tr>
<tr>
<td>Change Resistance</td>
<td>Waterfall is often mistakenly preferred from a desire for risk aversion, making Agile transformation difficult [2], [5]. Part of this challenge is due to high turnover both in the government and contractor roles.</td>
</tr>
<tr>
<td>Non-Agile Functions</td>
<td>Interfacing with non-Agile teams within the organization can hinder Agile team progress [2], [39]. ATO’s for example caused multiple organizations issues.</td>
</tr>
<tr>
<td>Large-Scale Agile</td>
<td>Bureaucracy and scaling can be a significant challenge. This can make it harder to get developers and the end-user communicating.</td>
</tr>
<tr>
<td>Testing</td>
<td>Current architecture with separate, lengthy developmental and operational testing does not match Agile. There is significant resistance and change that needs to happen for the USAF testing community to be truly Agile.</td>
</tr>
</tbody>
</table>

4.2 Potential Ways to Overcome Challenges

Despite the many challenges listed above, the USAF is continuing to move forward with the Agile implementation. The US federal government used to have more agile characteristics before Waterfall was mandated in the 1980’s. Therefore, according to Schoeni, the USAF can become more agile like again [5]. The DSB recommends iterative development practices for
future success [3]. Decisions made early have far reaching effects including end-of-life, especially in software [3]. These early decisions also effect the ability to adjust to changing requirements [3]. Just as the USAF sought flexibility and agility with lean and even earlier predecessors, the USAF is always seeking to improve its acquisition process [32]. It is also always seeking to find transformational technologies [34]. This means that innovation will need to be fostered in USAF employees [34]. This is a radical change, requiring that ideas are more important than rank, being wrong is not a failure, a culture of sharing ideas [34]. However, innovations occur more quickly at companies that successfully create an Agile culture [11]. These desired traits for innovation that are desired by the USAF line-up with Agile principles.

However, to make this a possibility all leadership will have to learn how Agile truly functions [11]. It is not anarchy or command and control but with faster results [11]. Below are ways in which the USAF can work towards becoming more Agile.

4.2.1 Laws, Regulations and Guidance

The DSB also is very concerned that the US Government does not have the expertise for Agile and recommends Congressional intervention [3]. In contrast others say that better regulations is not the solution [5]. Schoeni recommends three things: processes for IT outside of the FAR, changing regulations that do not align with Agile processes, and cutbacks in Congressional review of IT programs [5]. He recommends considering a replacement for the FAR since it is over 2000 pages [5]. If rewriting the FAR is done instead the bias for Waterfall needs to be written out [5]. For example – deleting all references to large milestone requirements [5]. An interviewee offered an interesting perspective, that perhaps the most Agile action with regulations was not to rewrite regulations to match Agile but rather permanently fix regulations to be process neutral. Lt. Gen. Bunch stated that upcoming legislation, such as the National Defense Authorization Act should help cut down on bureaucracy that hinders processes [29]. The National Defense Authorization Act should also encourage quicker adoption of new technologies [29]. Until then, Bunch is recommending that government employees keep in mind Congress’s intent and utilize existing means [29]. Regardless, according to a report from the DSB on the state of the DoD acquisitions, they stated that a process should be in place through policies that enable increments of functionality to be delivered in under 18 months [31].
One interviewee mentioned that they are trying to gain a support for an IT development color of money. This would address the confusion on whether sustainment or development funds should be utilized once a minimally viable product is fielded.

### 4.2.2 Contracting

The DSB report had a multitude of recommendations related to contracting. The DSB task force found that contracting best practices needs to be adjusted to properly incentivize Agile implementations properly [3]. They recommended “contracting software development as a service”, “paying for the overall outcome as a service” and “fixed-price development programs where hybrid iterative approaches adopted by industry to control costs” [3]. The DSB report recommends ensuring contracts are made for sustainment since software never dies [3]. One interviewee mentioned that in order for prime contractor use to be successful with an Agile project the government needs to ensure they have enough of the right people to be embedded in the Agile processes. This interviewee found that level of effort was an effective contract method. Another interviewee had great success with contracting plus and even found that contracting was not problematic in their Agile transformation.

According to DSB report in 2018 there are key areas to focus on, primary is the software factory [3]. This can be utilized as an evaluation criteria in source selection [3]. For example, one organization found a way to provide internal cost data software which gave insightful information [3]. They also recommend that a standardized list of criterion is developed to review software factories to determine minimally viable proposals [3]. The DSB recommends that DoD and partners continue to adopt best practices for Agile utilizing minimally viable products [3]. They also recommend that cost estimates should be updated to SLOC (source lines of code) approach and utilize WBS (work breakdown schedule) data with contractors [3]. The DSB also suggests using a new framework for status estimation such as velocity or control charts [3].

Regardless the DoD needs to change contracting to encourage traditional defense contractors to utilize Agile implementation on contracts [3]. This is necessary since in a good Agile implementation companies can have immense trust in each other [17]. For example, at Toyota their suppliers give proprietary information to Toyota knowing that this will not result in an appeal for discounts [17]. Toyota is aware that there are greater advantages in the long term by
not utilizing short term gains [17]. To accomplish this win-win agreements are fundamental in fostering collaboration [17]. An alternative would be a target-cost where both parties agree on a target-cost that includes 10-20% on the base price for all changes [17]. If the target-cost cannot be met both parties enter negotiations [17]. Most of the time the cost bearer should be the DoD [17]. This method has successfully worked for Toyota and their suppliers [17]. This process is very different than fixed-price with strict change procedures [17]. One success story of mutual trust came from an interviewee. Although it was not possible to put turnover in the contract the contracting officer created a successful relationship where they were able to keep 10% or less turn over on the prime contractor’s development team. Another interviewee noted that unlike Waterfall methods, under Agile, they were able to align competing contracting company needs so that contractors from competing firms worked well together.

Finally, there could be a need for a completely role in the government team of an end-user contracting representative [20]. Although not a replacement for the actual end-users this person would have a unique role. Not only would they represent the end-users but they would also have the technical authority (per the Procurement Contracting Officer) to command contractor actions within limits [20]. This will be a difficult role to staff since it would require understanding of best acquisition practices and Agile development [20]. However, this could help mitigate the risks associated with a looser contract by having direct contact with users but also regulating it. This role would have a direct influence on contractor actions and better empower end-users who have limited to no acquisitions training or knowledge.

4.2.3 Learning Curve

The 2018 DSB study also had some recommendations related to the learning curve challenges.

“The study recommends DoD adopt best practices on risk reduction and metrics in formal program acquisition strategies [3].”

The DSB recommends integrating these into current and legacy programs at all phases [3]. In order to accomplish this they recommend greater software expertise in the employees [3]. This means creating trained Agile experts to lead others in code development in additional to program managers [3]. They are also calling for prime defense contractors to do the same [3]. Schoeni agrees that one of the keys to success is empowering the DoD employees through coaching and
training [5]. An interviewee echoed that hands-on quality training empowered their team to succeed with Agile implementation. It is also extremely important to have products owners engaged and trained early in the process – they were found to be a key factor of Agile implementation success [2]. It is also important to note that Agile teams should not be put into motion until they are prepared [4]. This means the correct project and sufficient Agile skills mastered [4]. Having an Agile promoter within the government team that is developing a product and authorized to work with the contractor and government team is seen as a best practice [20]. The USAF also is planning on creating venues for Airmen from different areas to collaborate with self-organizing subject matter experts [34].

4.2.4 Waterfall Remnants

The DSB recommend that all ACAT I, II, and III programs entering milestone B need to implement Agile processes, but also recommend that Agile processes are not over described to allow for adaptability [3]. They recommend that development programs at a minimum switch to a hybrid Agile-Waterfall approach and legacy programs to be evaluated to determine if a change in methods makes sense [3]. Another way to help enforce a more Agile culture would be for most programs to have the expectation to deliver actual functionality within five years [6]. This would not only increase warfighter benefit but decrease plans for inventing on a schedule [6].

“The study recommends DoD adopt best practices on risk reduction and metrics in formal program acquisition strategies [3].”

One interviewee commented on the challenge of using quality metrics. Rather than focusing on scope, their team found customer satisfaction to be a quality metric of success. The DSB recommends incorporating these into current and legacy programs at all phases [3].

In “The Challenges of Being Agile in the DoD”, Broadus notes the challenge with Agile documentation not meeting typical milestone expectations [20]. Broadus recommends using a military analogy of “Commander’s Intent” [20]. Adapting a plan if the actuality veers from the original intention [20]. To do this Broadus recommends that during contracting negotiations and tailoring that there is a focus on how documentation is required due to DoD policies that are not productive towards Agile [20]. Although, clearly still a work around, since documentation is being developed that is not utilized – it is a step forward.
In addition, the USAF sees value in directly tackling the hierarchal military compared to Agile’s equality standards [34]. In its Strategic Master Plan, it states that although the chain of command will always been needed there will also be further attempts taken to empower people at lower levels [34]. It also calls for a renovation of processes [34]. This will be done by continuing to encourage continual improvements, reeducation of oversight and redundancies and more tailored approaches to reporting [34]. An important perspective to keep in mind is that although high up managers may at first feel a loss of control – they have now gained their valuable time back to do things that only they can do [11]. For example, organizational vision, focusing work, removing obstacles and putting the right people in the right job [11].

4.2.5 Change Resistance

The DSB task force recommends fitting Agile to a particular program [3]. One example is a mix process which has an unchanged end-product but uses a “small-scale major development program [3].” Another example is for a current major development program to do a large-scale Agile implementation [3]. However, they stated the law requires a Configuration Steering Board that involves high level approval processes [3]. If a program is new they recommend considering a service model where “best” value” is the selection criteria for contractors [3]. Finally, they mention Legacy programs could utilize Agile to find errors sooner by having sprints between milestones, although it does not shorten the project schedule [3].

When transforming an enterprise, it is ill-advised to take procedures from one organization to another. Instead, the values behind them is what should be used to create new procedures for the next organization [17]. The order of the transformation is very important [4].

“The most important – and the most frequently overlooked – are the pain pints felt by customers and employee on the one hand and the organizations capabilities and constraints on the other [4].”

Some companies did a 100% complete switch at once that succeeded, demonstrating the power of Agile [2], [4]. This needs a lot of buy-in, which requires a complete change of mindset for employees and is high risk [4].
With an Agile transformation it is important to realize that cost rewards are usually delayed – up to 7 years [4]. However, things like higher quality products should be immediate [4]. It is recommended for success to do a gradual, continuous implementation of 5-10 teams a month for many years [4]. True reforms should be done gradually and deliberately [5]. Each implementation should be seen as a mini-achievement [2]. The transformation to Agile should follow an Agile process[4]. Employees can become demoralized so it is important to have reminders that issues are not due to Agile but instead the organization [2]. Start with the highest potential rewards teams then learn and follow in a series from there [4]. Change leaders should also be selected for the teams [2]. New people are often best since they lack a history with the enterprise [2]. Finally, communication about progress was a key for many case studies in Agile transformation [2]. Openness about progress and project status also helped [2].

To help address change resistance at the lower levels there needs to be a change in culture [34]. The USAF Strategic Plan states that there needs to be a culture of new ideas, concepts and openness [34]. Not only do new ideas need to be encouraged, there needs to be understanding that new ideas will likely fail, and even if they do not it will seem to be worse than the current status quo [34]. However, out of every thousand ideas there could be a transformative concept [34]. There needs to be an understanding that before success there will be multiple failures [34]. A culture of risk avoidance should change to one of stagnant avoidance [34]. This should be driven forward by leadership [34].

Although counter to how the military typically runs its organizations, it may be worthwhile to consider less rotation through Agile teams [11]. This will give Agile teams time to mature and be able to customize their Agile practice well [11]. Additionally, according to Rigby et al., stable teams are 60% more productive and responsive to customers [11].

4.2.6 Non-Agile Functions

Engaging the entire organization secures stakeholder buy-in is critical to Agile transformation [2]. Additionally, using an Agile mindset to create a backlog of opportunities that are broken into pieces with limited overlap was another critical piece [4]. This makes it simpler to see gaps and encourages discovery [4]. One interviewee felt that non-Agile functions could use education especially considering that with Agile there is no one size fits all. Rather, they emphasized
Agile, no matter your role, requires critical thinking to determine what makes sense for the given situation.

There was one interviewee whose team was working through challenges with the cybersecurity ATO function. They were working with the cybersecurity functions to validate their automated, daily testing that checked many of the concerns that the ATO representatives had. They had worked towards using their Agile processes by involving the cybersecurity team to ensure that they were meeting the intent of the current ATO process through a different means. This could prove to be successful for other Agile programs as well.

Johnson encourages that organizations that provide “unnecessary oversight” or “funding pass-through” to be phased out [30]. This could empower the program manager to make more decisions and increase productivity [30]. This is also encouraged in the Air Force Strategic Master Plan, where it is stated that one of the tasks to make the USAF more Agile is to find way to speed up feedback loops of information [34]. It is stated that is should not only be done up the chain of command but also laterally [34].

It has also been found to be beneficial-for leadership to become Agile, although not all roles will be suited to this [11]. When this happens organically it can smooth out the implementation process, making management more of an alley and freeing up management’s time [11]. Utilizing backlogs to handle and divide leadership issues has worked well for many companies [11]. It can also be an effective way of communicating with employees about what it is that management does [11]. Finally, it can help align different areas at a high level [11]. This can actually make Agile team implementation with non-Agile function teams better [11].

4.2.7 Large-Scale Agile Needs

Large implementations struggled if they did not have automated testing [2]. This reflects the DSB finding that the USAF should develop better machine learning talent for verification and validation [3]. One interviewee commented that within their organization they had successfully scaled Agile by engaging with the teams utilizing active coaching and refining Agile as they went. Implementing Agile by starting small and letting the word spread is a great way to implement Agile for large scale [11]. Agile then is able to spread as each success creates another individual who is excited to tell their peers about Agile [11]. Another recommendation is to
utilize a “block program” approach where single pieces of functionality are built at a time instead of trying to meet all specifications at once [6].

4.2.8 Testing

Although traditional defense testing and Agile utilize significantly different methods Broadus noted that these differences do not mean that independent team testing is not needed to determine the effectiveness and the appropriateness completed during Operational Testing [20]. In contrast, an interviewee felt that testing members could be part of the team, but have a separate chain of command to keep the intent of independence while integrating with Agile methods. Broadus emphasizes that the continuous testing reduces risk by finding issues earlier in development [20]. An interviewee also noted that daily automatic testing is a great step towards modifying current USAF testing. This interviewee recommended less of the standard testing processes and accepting a little bit more risk.

A leadership intervention may be required for a short term fix [11]. If a project is truly a priority then it should be prioritized for the testing organizations as well, even if they are not Agile [11]. Testing roles need to change [11]. Also by engaging with testing functions differently, such as asking them how to do what you need done is better than dictating it [11].

4.2.9 Summary of Challenge Mitigations

Although there is clearly no magic bullet to the challenges being experienced during the USAF Agile transformation, there still are a multitude of ways to address the challenges at different levels. Below, in Table 4-2, is the summary of the challenge mitigation strategies uncovered.
<table>
<thead>
<tr>
<th>Challenge Category</th>
<th>Mitigation Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Law</td>
<td>Update regulations to reflect Agile in software and decrease Congressional reviews [5]. Engage to educate Congress and/or staffers on new Agile metrics such as burn down charts that will be utilized. A new color of IT money would be extremely beneficial.</td>
</tr>
<tr>
<td>Contracting</td>
<td>Update contracting standards to meet Agile needs [3]. This will take a concerted effort to gain trust between government and contractors [17]. An end-user contracting representative could be a way to handle Agile challenges in contracting [20].</td>
</tr>
<tr>
<td>Learning Curve</td>
<td>Coaching and training will be needed [5]. While the DoD tends to excel at training, active coaching is a newer concept in acquisitions.</td>
</tr>
<tr>
<td>Waterfall Remnants</td>
<td>Continue to educate high up leadership on new Agile metrics and the problems of utilizing Waterfall metrics of scope, schedule and cost. Continue to empower employees and lower levels.</td>
</tr>
<tr>
<td>Change Resistance</td>
<td>This is best accommodated by doing a slow roll-out with a few teams at a time [2], [4]. It will also require adjustments to rewards for team based behavior and strategically bringing in new individuals [4]. Minimize turnover on Agile teams.</td>
</tr>
<tr>
<td>Non-Agile Functions</td>
<td>Stakeholder buy-in from all areas of the enterprise is needed [2]. This is a culture shift that needs to occur [4]. Engaging in an open way with these areas eases transition. Unnecessary oversight and pass-throughs need to be eliminated [30]. Encourage some high leadership to make themselves Agile [11].</td>
</tr>
<tr>
<td>Large-Scale Agile Needs</td>
<td>Automated testing is needed for large companies to handle the amount and frequency of testing needed [2], [3]. Slow roll outs are necessary to ease into change.</td>
</tr>
<tr>
<td>Testing</td>
<td>Work to educate and collaborate with testing at all levels. Eventually testing will need to be reorganized to be better integrated into the development process.</td>
</tr>
</tbody>
</table>
4.3 SWOT Analysis

A common and useful tool to arrange results of an analysis is SWOT, Strengths, Weaknesses, Opportunities and Threats [8].

“A SWOT analysis involves identifying the factors, both external and internal, that are either favorable or unfavorable to achieving a given objective.” [8]

Strengths are anything that enable the USAF to have an edge in achieving an Agile transformation. Weaknesses are anything that could cause an Agile transformation to fail [8]. Opportunities are any chance to flourish in the USAF’s broader environment [8]. Finally, threats are something in the broader environment that could cause problems for the USAF [8]. This tool was utilized to gather the findings for the current state of transformation and bring together the earlier findings.

4.3.1 Strengths

The United States military is often listed towards the top of world military powers. The USAF ranks itself as a leader in air power in the world. Lt. Gen Bunch stated that the USAF has deployed air superiority using methods that no one could even imagine, stating that no one in the world has caught up to USAF air power [29]. Being one of the leading world military powers gives the USAF more resources at its disposal. Despite its challenges the US continues to be at the leading edge of warfare technologies and often collaborates with other world powers. Finally, the USAF has a very large workforce which has in-depth weapon knowledge. All of these characteristics give the USAF an edge at being successful at an Agile transformation.

4.3.2 Weaknesses

Unfortunately, the down-side to some of the USAF’s strengths are that because of its size and the amount of bureaucracy it is slow to change. In order to elicit a cultural change, there is a great inertia to overcome. As one of the interviewees noted, Waterfall tools such as earned value management are no longer needed. This can exacerbate the time it takes for this large organization to change. Also, within the USAF there is a lack of adequate number of employees that have systems engineering or Agile software development skills. This is worsened by poor retention and hiring practices for the needed talent. There is a growing concern among
leadership, within the military and also the larger government that the US may be losing its edge in military warfare [31]. There also is a tendency of the USAF acquisitions being scared of failure, demonstrated by holding onto previous mistakes. All of these skills are essential for a successful transformation.

4.3.3 Opportunities

Despite the weaknesses mentioned above, there are still numerous opportunities in the USAF ecosystem. Most modern weapon systems are software dependent – meaning that a software focused development would be beneficial. Succeeding with weapon development not only makes the US position stronger but makes the adversaries position weaker [6]. Along with this there is growing attention on Agile as a software-focused development and its many benefits. Agile is creates many opportunities for the USAF. According to one interviewee it creates a better alignment of stakeholder needs noting that teams using Agile are more united and experience less conflict. Broadus mentions this as well, that there is an alignment of the user needs to the work being accomplished [20]. As mentioned throughout this paper and by one of the interviewees, the biggest opportunity presented by Agile is the possibility of not only delivering more capability at a higher quality but also delivering it in less time. This is probably the most focused on opportunity, especially considering Waterfall cultural focus on scope, schedule and cost.

4.3.4 Threats

As mentioned above, culturally, even those outside of the USAF still have an emphasis on cost, schedule and scope, and Congress is no exception to that. Considering the amount of influence Congress has, in addition to the fact that Congress does not get any value directly from an acquisition program office, there is a communication challenge between the USAF and Congress. The USAF is, of course, first a military organization – meaning that hierarchical control is emphasized along with command and control. This combined with Waterfall cultural creates a misalignment between the USAF culture and Agile culture. There is certainly no shortage of challenges to overcome for the USAF to adapt Agile [20]. This makes the idea of true cultural change for Agile threatened.
4.3.5 SWOT Analysis Summary

There are a multitude of strengths, weaknesses, opportunities and threats to consider when analyzing the USAF Agile transformation. While a USAF Agile transformation is a grand endeavor to take on it has probable great success for future weapon system development [20].

The information discussed above is summarized below in Table 4-3.

Table 4-3 Summary of SWOT Analysis

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>• One of, if not, the world military power</td>
<td>• Slow to change, lots of inertia to overcome</td>
</tr>
<tr>
<td>• Leading in military warfare technologies</td>
<td>• Lack of know-how with systems</td>
</tr>
<tr>
<td>• Large workforce with weapon system knowledge</td>
<td>engineering and Agile software development</td>
</tr>
<tr>
<td></td>
<td>• Talent retention &amp; hiring processes are</td>
</tr>
<tr>
<td></td>
<td>ineffective at obtaining the right talent</td>
</tr>
<tr>
<td></td>
<td>• USAF is scared of failure, it holds onto</td>
</tr>
<tr>
<td></td>
<td>mistakes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunities</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Majority of weapon systems are software</td>
<td>• Communication challenges between</td>
</tr>
<tr>
<td>dependent</td>
<td>Congress’s and the USAF</td>
</tr>
<tr>
<td>• Growing attention to Agile and its benefits</td>
<td>• Misalignment between USAF culture and</td>
</tr>
<tr>
<td>• Better alignment of stakeholder needs</td>
<td>Agile culture</td>
</tr>
<tr>
<td>• Delivering more capability at a higher</td>
<td></td>
</tr>
<tr>
<td>quality to the warfighter in less time</td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

"Why is it that so many veteran [government] administrators develop their incisive perception and reform proposals when they are no longer in a position to act to improve the situation?" [1]

Gene Zuckert, former Secretary of the Air Force

In some sense this quote captures the inspiration for the higher goal of this thesis. In other words, the higher goal is to bring together a literature review of academic, industry and government papers to create a holistic view of the USAF Agile Transformation to empower current government employees to make this transformation successful. To accomplish this goal this thesis began with an extensive review of current literature on Agile, discussing what makes it so different from Waterfall and even some examples of Agile in the USAF. This was followed by an analysis of the current ecosystem of the USAF as an enterprise. This involved finding the major ecosystem factors along with a stakeholder analysis. In order to better understand the USAF Agile transformation, an analysis of its current challenges and mitigations was generated and analyzed. A SWOT analysis was then completed to bring together the expansive findings from a holistic analysis. The USAF enterprise is both a complex and complicated system warranting a holistic analysis. The pure volume of information can be overwhelming and challenging to synthesize. The insights and findings discussed next are offered to help empower USAF employees to make more educated decisions.

5.1 Insights and Findings

The primary contributions in this thesis are the synthesized holistic analysis, summarized in Table 3-1, Figure 3.4, Table 4-1, Table 4-2 and Table 4-3. It is important to keep in mind that these are “models” in the sense that they are simplified versions of the real system. Models add value because these simplified versions of the real thing, make it easier for human comprehension. The first finding was that political, regulations, resource, and technology ecosystem factors all have, or have the potential to have a major impact on the USAF Agile transformation. This finding is so impactful because most articles focus on one or a couple of the factors – but none talked about all four. Second, the stakeholder value network created a unique view of the relationships that impact an acquisitions program. This highlights a few stakeholders, some of whom were unexpected to be a focal point. A focus of many defense
pieces, congressmen are highlighted as a potential problematic stakeholder since the acquisition program needs them, but they do not receive any value directly from the acquisition program. Overlooked in defense pieces but often discussed in Agile literature are the non-Agile functional areas. These stakeholders also receive limited, if any, value from the acquisition program but the acquisition program needs them. In order for the USAF Agile transformation to be successful buy-in from non-Agile functional areas will vital. Finally, a stakeholder not mentioned in literature, but identified from interviews, is the warfighter lead command. This stakeholder is also vital to enable acquisition program personnel to reach the true end-user to be able to fully embrace Agile technology. If they fail to see the value that an airman working day-to-day with the weapon system can add in comparison to their identified representative they are unlikely to facilitate.

Third, the challenges and mitigations re-emphasized the importance of non-agile functional areas, particularly contracting and testing. It also focuses on a new point – a cultural change from Waterfall. Active coaching, mentioned frequently in literature was found to be utilized in successful USAF Agile programs, although the interviewees are not familiar with the term. The challenges and mitigations also highlight actions that can be taken at different levels to move towards an Agile transformation. For example, having a new color of money for IT requires congressional approval and support from high-level government officials. In contrast, an individual such as a single contracting officer embracing critical thinking can utilize a method such as contracting plus to help enable an acquisition program to become more Agile.

Finally, the SWOT analysis shows the USAF has a multitude of strengths that unfortunately also created weaknesses. However, despite some threats there is also a list of opportunities, some of which are the inspiration for the transformation. Although perhaps not the ideal scenario of near zero negatives of a transformation, the SWOT makes it clear that the USAF must press forward with the Agile transformation. To fail to do so risks losing their ranking as a lead world military power, a risk that the USAF would likely classify as too great.

5.2 Future Research

Despite the encouraging findings in this research, this thesis is an initial exploration only. Although this thesis utilized a systems approach for a holistic view of the problem space, it is impossible to truly encompass everything. Additionally, due to time limits, further validation of
results was not possible, future research is vital to do this. A case study of USAF Agile programs is recommended to more rigorously determine if the literature review findings are reflective of the actual USAF Agile Transformation. A case study focusing on large acquisitions programs trying to make their software portions Agile would be extremely beneficial. Although many small USAF programs have attempted Agile, there are few, if any, large programs successfully implementing Agile. Another limitation of this research was treating Agile and DevOps as the same for USAF transformation purposes. In addition to Agile and DevOps having distinctly different backgrounds, there is also Agile DevOps a combination of the two [3]. In future research, there is a need for greater differentiation between them and how that impacts implementation efforts in any industry.
6 References


Acronyms and Abbreviations

ACAT – Acquisition Category
AFB – Air Force Base
ATO – Authority to Operate
DAU – Defense Acquisition University
DoD – Department of Defense
DoDI – Department of Defense Instruction
DSB – Defense Science Board
FAR – Federal Acquisition Regulation
FY – Fiscal Year
HR – Human Relations
NSA – National Security Agency
OT&E – Operational Test and Evaluation
PDCA – Plan-Do-Check-Act
PEO – Program Executive Officer
ROI – Return on Investment
SLOC – Source Lines of Code
SWOT – Strengths, Weaknesses, Opportunities, Threats
US – United States
USAF – United States Air Force
WBS – Work Breakdown Schedule
XP – eXtreme Programming
Appendix A: Defense Acquisition Life Cycle Compliance Baseline (Pre-Tailoring)
Tailoring and Program Models - The models provide baseline approaches. A specific program should be tailored to the unique character of the product being acquired. 

Alternative Pathways - Today, the Department predominantly uses two acquisition pathways that are well understood and often regular, one for Major Defense Acquisition Programs (MDAPs) (describe in DoDI 5000.2), and one for Urgent/Emerging Operational Needs (U/EON) (describe in DoDI 5000.7). However, in the dynamic and ever-evolving environment, DoD needs alternative pathways to acquire capabilities faster than these two models alone can support. The FY 2016 NDAA provided such an additional pathway, referred to as "middle tier acquisition." This provision recognizes DoD's need to move faster on compelling technologies that are too early to declare as an acquisition program, but have the ability to provide significant warfighter advantages if delivered faster.
Appendix B: MIT COUHES Approval

Consent to Participate in Interview
A Systems view of Agile & DevOps in the Air Force: An Investigation of the Challenges of Implementation

You have been asked to participate in a research study conducted by Elizabeth Bieler and Donna Rhodes from the Massachusetts Institute of Technology (MIT). The purpose of the study is to leverage personal experience with Agile and DevOps in order to understand implementation challenges within Air Force Acquisitions. The results of this study will be included in Elizabeth Bieler’s Master’s thesis. You were selected as a possible participant in this study because of the experience you have as an engineer or program manager with an Agile or DevOps program which we believe may offer valuable insight for this research project. You should read the information below, and ask questions about anything you do not understand, before deciding whether or not to participate.

- This interview is voluntary. You have the right not to answer any questions, and to stop the interview at any time or for any reason. We expect the interview to take 45-60 minutes.
- You will not be compensated for this interview.
- Unless you give us permission to use your name, title and/or quote you in any publications that may result from this research, the information you tell us will be confidential.
- We would like to record this interview so that we can use it for reference while proceeding with this study. We will not record this interview without your permission. If you do grant permission for this conversation to be recorded, you have the right to revoke recording permission and/or end the interview at any time.

This project will be completed by 10 Aug 2018. All interview recordings will be stored on a password protected external hard drive that is kept in a locked cabinet until a year after that date. The tapes will then be destroyed.

I understand the procedures described above. My questions have been answered to my satisfaction, and I agree to participate in this study. I have been given a copy of this form.

(Please check all that apply)

- I give permission for this interview to be recorded.
- I give permission for the following information to be included in publications resulting from this study:
  - [ ] my name  [ ] my title  [ ] direct quotes from this interview

Name of subject

Signature of Subject __________________________ Date __________

Signature of Investigator ______________________ Date __________

Please contact Elizabeth Bieler by email at ebieler@mit.edu or by phone at (937)-367-9118, or Donna Rhodes by email at rhodes@mit.edu or by phone at (617)-324-0473 with any questions or concerns.

If you feel you have been treated unfairly, or you have questions regarding your rights as a research subject, you may contact the Chairman of the Committee on the Use of Humans as Experimental Subjects, M.I.T., Room E25-145b, 77 Massachusetts Ave, MA 02139, phone 1-617-253-6787.

EXEMPTION GRANTED ON 26-Apr-2018
To: Elizabeth Bieles
From: Leigh Finn, COUHES
Date: 04/26/2018
Committee Action: Exemption Granted
Committee Action Date: 04/26/2018
COUHES Protocol #: 1803282170
Study Title: A Systems view of Agile & DevOps in the Air Force: An Investigation of the Challenges of Implementation

The above-referenced protocol is considered exempt after review by the Committee on the Use of Humans as Experimental Subjects pursuant to Federal regulations, 45 CFR Part 46.101(b)(2).

This part of the federal regulations requires that the information be recorded by investigators in such a manner that subjects cannot be identified, directly or through identifiers linked to the subjects. It is necessary that the information obtained not be such that if disclosed outside the research, it could reasonably place the subjects at risk of criminal or civil liability, or be damaging to the subjects’ financial standing, employability, or reputation.

If the research involves collaboration with another institution, then the research cannot commence until COUHES receives written notification of approval from the collaborating institution’s IRB.

Unless informed consent is waived by the IRB, use only the most recent, IRB approved and stamped copies of the consent form(s).

Adverse Events: Any serious or unexpected adverse event must be reported to COUHES within 48 hours. All other adverse events should be reported in writing within 10 working days.

Amendments: Any changes to the protocol, including changes in experimental design, equipment, personnel or funding, must be approved by COUHES before they can be initiated, except when necessary to eliminate apparent immediate hazards to the subject.

Human subjects training is required for all study personnel and must be updated every 3 years.

You must maintain a research file for at least 3 years after completion of the study. This file should include all correspondence with COUHES, original signed consent forms, and study data.