Using System Dynamics Modeling to Understand the Impact of Federal Constraints in Implementing the Agile Methodology within the U.S. Federal Government

by Roderick D. Capili

B.S. in Aerospace Engineering, May 1995, San Jose State University
M.S. in Management, June 2009, Regis University
Master of Systems Engineering, March 2015, Embry Riddle Aeronautical University

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of the George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

January 19, 2018

Praxis directed by

Gina Guillaume-Joseph
Professorial Lecturer of Engineering Management and Systems Engineering

Alan Gideon
Professorial Lecturer of Engineering Management and Systems Engineering

The School Of Engineering and Applied Science of The George Washington University certifies that Roderick D. Capili has passed the Final Examination for the degree of Doctor of Engineering as of 19 Dec 2017. This is the final and approved form of the praxis.

Using System Dynamics Modeling to Understand the Impact of Federal Constraints in Implementing the Agile Methodology within the U.S. Federal Government

Roderick D. Capili

Praxis Research Committee:

Gina Guillaume-Joseph, Professorial Lecturer of Engineering Management and Systems Engineering Praxis Co-Director

Alan Gideon, Praxis Professorial Lecturer of Engineering Management and Systems Engineering Co-Director

E. Lile Murphree, Professor Emeritus of Engineering Management and Systems Engineering, Committee Member

Amir Etemadi, Assistant Professor of Engineering and Applied Science, Committee Member

Mr. Bruce Farquhar, Senior Principal Systems Engineer, Orbital ATK, Committee Member

© Copyright 2018 by Roderick D. Capili
All rights reserved

Dedication

I dedicate this praxis to my wife Tracy and my daughter Leah. My constant companions in this great adventure of discovery and the never-ending quest to find the knowledge and answers that life has to offer.

Acknowledgement

I want to thank my family, Tracy, and Leah for your utmost patience and understanding on the many hours, days, weekends and holidays spent in completing this paper. I also want to thank Mr. Bruce Farquhar for lending his time and effort in reviewing and editing my work, as well as serving on my research committee. I also would like to extend my appreciation to Dr. Gina Guillaume-Joseph and Dr. Alan Gideon for providing guidance and direction in developing this research.

This paper is a culmination of a journey in learning, and I would be remiss not to address the influence and impact of my parents, Rene and Mila, my brother Remil and my sister Mirene, who in their own rights have given me the inspiration and fortitude to forge forward in life. To all my teachers and instructors that have served, mentored and inspired me in so many different ways, I can only wish I can be as capable and as influential as you have been to my existence, to the next generation of learners and achievers that I intend to serve.

Abstract

Using System Dynamics Modeling to Understand the Impact of Federal Constraints in Implementing the Agile Methodology within the U.S. Federal Government

The agile methodology is the incremental, continuous development and delivery of software, providing a continuous stream of software solutions that meet the customer's needs. The U.S. federal government has realized the advantages of this methodology and has decided to utilize this methodology in acquiring and developing software solutions for its agencies. However, the overall implementation of this methodology within the different agencies of the government has not been consistent both regarding its effectiveness and success; as such, the ability to deliver effective software solutions within the desired cost, schedule, and performance parameters has been a challenge. Using a system dynamics model, research was conducted to understand how constraints in requirements, schedule, contract types, performance capability, and information assurance have impacted the effective implementation of the agile methodology. The research results were validated against four agile projects, with the results showing cost constraints, schedule delays and performance decline when constraints are applied. Based on these findings, the extent and validity of the constraints' impacts in utilizing the agile methodology were proven to be significant factors, and understanding these impacts will help create plans and strategic direction on how agile can be implemented appropriately in federal agencies, advancing the understanding of general project management of agile projects within the U.S. federal environment.

Table of Contents

Dedication iv

Acknowledgement v

Abstract.. vi

List of Figures x

List of Tables xi

1 INTRODUCTION 1

1.1 Problem Discussion 2

1.2 Significance of Research 6

1.3 Justification of Research 7

1.4 Relevance and Importance 7

1.5 Purpose of the Study 7

2 LITERATURE REVIEW 9

2.1 What is System Dynamics and how can it be utilized? 9

2.2 System Dynamics and Software Development 12

2.3 The Agile Methodology Overview 14

2.4 The United States Federal Government and Agile Methodology 16

2.5 Challenges of Implementing Agile in the Federal Government 19

2.6 Requirements and how it relates to agile in the Federal Government 22

2.6.1 Role of information assurance in agile development 24

2.7 Contracts Management and Agile in the Federal Government 26
2.8 Role of Schedule Management and Agile in the Federal Government 28
2.9 Performance Management and Agile in the Federal Government 30
2.10 Contribution to Research 31
3 RESEARCH METHODOLOGY 34
3.1 Goals and Objectives 34
3.2 Scope and Limitations 34
3.3 Research Method 35
3.4 Base Model Overview 37
3.4.1 Defining the baseline 40
3.4.2 Factors for the baseline 41
3.4.3 System dynamics application 45
3.4.4 Baseline results and discussion 47
3.5 Case Study 1 – Purchase Care Program Integrity Enhancement Project 54
3.6 Case Study 2 – Online Payment Agreement (OPA) Project 55
3.7 Case Study 3 – Veterans Affairs (VA) Online Medication Reconciliation Project 56
3.8 Case Study 4 - Healthcare Associated Infections and Influenza Surveillance System (HAIISS) Phase II Project 57
4 RESULTS AND ANALYSIS 59

4.1 Case Study 1 Results – Purchase Care Program Integrity Enhancement Project 59

4.2 Case Study 2 Results – Online Payment Agreement (OPA) Project 62

4.3 Case Study 3 Results – Veterans Affairs (VA) Online Medication Reconciliation Project (JC3) 64

4.4 Case Study 4 Results – Healthcare Associated Infections and Influenza Surveillance System (HAIISS) Phase II Project 67

4.5 Interpretation of the Results 69

4.6 Utility of the Model 72

5 CONCLUSIONS AND RECOMMENDATIONS 74

5.1 Conclusions 74

5.2 Research Challenges 75

5.3 Recommendations for Future Research 76

References 78

Appendix A – Base Model Results 84

Appendix B – Case Study 1 Results 86

Appendix C – Case Study 2 Results 88

Appendix D – Case Study 3 Results 90

Appendix E – Case Study 4 Results 91

List of Figures

Figure 2-1: Example of software development framework and related activities (Rodrigues & Williams, 1997) 13

Figure 2-2: The agile methodology of software development (TechFAR Handbook, 2017) 15

Figure 2-3: Implementation model for the agile methodology in delivering software 33

Figure 3-1: Overview of the systems dynamics modeling approach (Martinez & Richardson, 2001) 36

Figure 3-2: The agile project dynamics model (APD) (Glaiel, 2012) 37

Figure 3-3: System dynamics baseline model for agile implementation 39

Figure 3-4: Interaction of critical factors for agile implementation (Cao et al., 2010) 43

Figure 3-5: System dynamics representation for study 45

Figure 3-6: Schedule results for base model testing 52

Figure 3-7: Cost results for base model testing 53

Figure 4-1: Cost results for Case Study 1 testing 61

Figure 4-2: Schedule results for Case Study 1 testing 61

Figure 4-3: Performance impact results for Case Study 2 63

Figure 4-4: Cost behavior results for Case Study 2 64

Figure 4-5: Performance reduction results for Case Study 3 65

Figure 4-6: Change in development rate results for Case Study 4 69

Figure 4-7: Cost behavior results for Case Study 4 69

List of Tables

Table 2-1: Benefits of implementing agile in the federal government (Miller, Hayes, and Wrubel, 2016) 18

Table 2-2: Challenges for implementing agile in the federal government (Miller, Hayes, and Wrubel, 2016) 19

Table 2-3: Contracts summary for implementing agile in the federal government (TechFAR, n.d.) 28

Table 3-1: Factors selected for baseline model 44

Table 3-2: Agile projects summary per contract type 48

Table 3-3: Baseline model results 50

Table 3-4: Case Study 1 project results 54

Table 3-5: Case Study 2 project results 55

Table 3-6: Case Study 3 project results 56

Table 3-7: Case Study 4 project results 57

Table 4-1: Cost and schedule variance for Case Study 1 60

Table 4-2: Cost and schedule variance for Case Study 2 63

Table 4-3: Case Study 3 validation results 67

Table 4-4: Case Study 4 validation results 68

1 INTRODUCTION

In the federal government, the adoption of the agile methodology for software acquisition and development continues to grow. Vechnicki and Kelkar (2017) recently discussed that over 80% of major information technology investments from reporting federal agencies in 2017 had utilized the agile approach in their software development and acquisition. Basing their assessment on collected data from the federal reporting ITDashboard.gov website, Vechnicki and Kelkar (2017) reported that since 2012, the use of agile in federal agencies[1] has shown a steady increasing trend from roughly 18% of all reported major information technology projects in the federal government to above 80% in the span of 5 years. Ironically, as the adoption rates for agile continue to grow and more federal agencies become familiar with the methodology, Vechnicki and Kelkar (2017) also noted that the relative impact of using agile has not translated into a clear advantage—in cost or schedule—over the more traditional sequential approach of using the waterfall methodology in software development; they stated that this is because of the varying reported difficulties in managing the projects that utilize agile, as well as the differing viewpoints regarding what federal projects are more suited to an agile approach.

Although the use of the agile methodology has not been directed or established as the only path forward for all federal software acquisition and development, the perceived advantages of using this methodology—with the iterative delivery of software capability to the consumers—has captured the attention of federal leadership in their desire to improve software development and acquisition. As such, guidance has been provided

[1] For the purposes of the current study, "federal agency (ies)" and "agency" refers to the United States federal government as a whole, with specific agencies identified when needed.

from the top levels of federal leadership "to take clear considerations for the use of agile as means for software development methodology" (TechFAR, n.d.), in seeing how "agile provides the flexibility and the means to adapt to changing needs" suited for addressing the evolving 21st century needs of the federal government (TechFAR, 2017).

1.1 Problem Discussion

In July 2012, the United States Government Accountability Office (GAO, 2012) released a report that identified the challenges in implementing software development using the agile[2,3] methodology and effective practices that can potentially increase the implementation and success rate of agile adoption for federal agencies. The report identified 14 challenges with adapting and applying agile in the federal environment and recommended 32 practices and approaches that align with five key project management activities namely:

1) Strategic planning

2) Organizational commitment and collaboration

3) Preparation

4) Execution

5) Evaluation

Some of the "best" practices proposed include continuous stakeholder/feedback engagement to minimize the challenge, continuing assessment of requirements, proper

[2] For the purposes of the current study, "agile" and "agile methodology" refers to any of several software development practices that use a collaborative, iterative development environment in delivering incremental software solutions for the customer.
[3] Note: "agile" and "agile methodology" are used interchangeably throughout this research.

oversight and management for performing teams, and application of correct contract types for execution of projects. According to the report, adoption of these strategies will help facilitate a common understanding of the strength and weaknesses in implementing the agile methodology and can help an overall development management strategy for leaders and decision makers (GAO, 2012).

On 1 October 2013, the Centers for Medicare and Medicaid Services (CMS), an organization within the Department of Health and Human Services (HHS), launched the healthcare.gov website to provide a federal insurance marketplace in support of the Patient Protection and Affordable Care Act. Within 24 hours of the website's launch, the website was quickly overwhelmed by users. The federal government conceded that this effort to develop a portal for users to access the federal insurance marketplace was broken, and additional development and work effort was noted as being needed to fix the system (Timeline: US healthcare law's technology breakdown, 2013). On July 2014, the United States Government Accountability Office (2014) released a report to Congress, entitled "HEALTHCARE.GOV Ineffective Planning and Oversight Practices Underscore the Need for Improved Contract Management," which described how ineffective management practices, lack of understanding of the agile methodology, ineffective contract management, and unrealistic schedule and performance expectations significantly contributed to the failure of the healthcare.gov website's launch, costing millions of dollars in taxpayer money during and after the release of the website. In utilizing the agile methodology, the GAO report (2014) concluded that multiple lack of understanding of the implications of the changing requirements while pursuing a compressed timeline to release the software were the significant factors in the limited

performance capability of the website, thereby ensuring a failed launched once it was released for consumer use. In addition, the GAO report (2014) also discussed the inefficient management of contract expectations: although funds were increased to hopefully alleviate some of the agile challenges that existed during the development process, failure to understand that software re-work and poor quality of software development impact to an agile effort ultimately resulted in software delivery that was flawed and did not meet customer expectations.

On the same day that the healthcare.gov website was launched, the State of Oregon tried to launch its own state initiative website, Cover Oregon, which was linked to the federal insurance marketplace for state-provided access, also in support of the Patient Protection and Affordable Care Act (PPAC). Like the healthcare.gov website, the Oregon state website also failed to launch and after numerous re-workings and multiple project delays, the State of Oregon conceded that their effort at the state level would not satisfy the provisions of the PPAC and on 25 April 2014 switched to the federal website healthcare.gov as the primary means of allowing Oregon users to access the newly formed insurance marketplace. With this decision to switch, it was determined that $305 million in federal taxpayer money was wasted in developing the Cover Oregon website (Committee on Oversight and Government Reform, 2016). In a report submitted to Congress by the Committee on Oversight and Government Reform (2016), it was further determined that for the Cover Oregon project, mismanagement and the lack of an overarching strategic software development plan—despite using emerging software development tools such as agile and an experienced software developer (Oracle)—poor

contract selection, and undefined requirements were significant causality factors in the failure of the project.

Despite the notable failure of the release of the healthcare.gov website and other major software-intensive applications sponsored by the federal and state governments, the adoption of the agile methodology to acquire or develop software for federal agencies has significantly increased by 60% of reported software development projects (Viechnicki and Kelkar, 2017). However, while the adoption rates for the agile methodology among federal agencies continue to grow, Reger (2017) noted that the trends in government reporting do not necessarily indicate that using the agile methodology also delivered a significant improvement over using the waterfall methodology. Based on reporting from the ITDashboard.gov website for major information technology (IT) investments, Reger (2017) showed that the advantages in product delivery or cost savings, the primary considerations for utilizing agile, were not significant enough to necessarily negate the utility of waterfall as a software development approach. Reger (2017) further noted that although agencies are set in their compliance with using agile as a means for software development, federal agencies continue to struggle with agile project management; often, federal agencies lack the understanding needed to truly manage the methodology in a federally constrained environment, negating the potential advantages of using agile versus the waterfall methodology.

As agile becomes more prevalent in federal agencies, the rationale for the utility of agile as a better alternative to the waterfall methodology will lose credibility if the perceived advantages in using this methodology, do not indicate a significant departure from previous development methods. Understanding the constraints that exist within the

federal environment and how these constraints can impact agile implementation will help define management decisions and actions when utilizing this methodology. Using system dynamics [4] principles and methodologies in assessing existing processes and implementation strategies within federal agencies, the goal of the present research is to provide federal program and project managers, as well as agile software developers, with an understanding of the different impacts of federal constraints when implementing agile.

1.2 Significance of Research

The TechFAR Handbook (2017) identified four key elements to consider when implementing agile in the federal government: requirements development, contract vehicles, pricing, and contract administration. These elements are important considerations because it defines the necessary scope of the agile effort and the basis for what the delivered product should be. As variables change and the development cycle matures, the basic scope for the effort becomes the deviation point of change for any cost, schedule, or performance constraint. As the utilization of agile for acquiring and developing software solutions continues to grow, the significance of understanding how changes in the development cycle impact the four elements takes an ever-increasing role to harness the advantages that the methodology aims to deliver. Although the TechFAR Handbook does not imply specific directives on how federal agencies should implement their agile efforts, the handbook notes that understanding these four key considerations, not only help agencies understand how agile is utilized, but also helps manage

[4] System dynamics is an approach to understand systems behavior through the modeling and simulation of the different variables that exist within the system.

expectations and management responsibilities in ensuring success not only for the federal agency, but also for the software developer.

1.3 Justification of Research

The research aims to advance the understanding of the factors that influence agile implementation within federal agencies. Using a systems dynamics model to validate values to these factors, the hope is that the resulting behavior and pattern that these factors exert will show management and decision makers how agile should be implemented within the constraints that these factors exert.

1.4 Relevance and Importance

This research is relevant for multiple reasons. First, as federal agencies continue to increase software spending and budget allocations, it becomes imperative that relevant value and utility are achieved in the acquisition and development of these software solutions. Second, as more and more agencies adopt agile as the preferred software acquisition and development methodology, understanding the different interactions that impact the proper management of an agile effort is critical to the overall execution of the project. Third, in applying systems dynamics modeling to an existing real-world system, the distinction between theoretical applications of engineering methodologies can gain a more practical utility when applied to solving real-world problems.

1.5 Purpose of the Study

The purpose is to validate the utility of system dynamics in understanding the different impacts and consequences of four key drivers (requirements development, contract vehicles, pricing, and contract administration) for implementing agile in federal agencies.

There are two key hypotheses presented for this research, as follows:

1. Null hypothesis: The implementation of the agile methodology in federal agencies is not impacted by federal acquisition regulations (FAR) for requirements, contract type, and schedule because of its iterative and continuous delivery of software solutions.

Alternative hypothesis: The agile methodology is impacted by federal acquisition regulations (FAR) for requirements, contract type, and schedules in its ability to provide software solutions that are responsive to customer needs.

2. Null hypothesis: The use of system dynamics is not an effective method for understanding the behavior of agile software implementation for federal organizations.

Alternative hypothesis: The use of system dynamics is an effective method for understanding the behavior of software implementation using metrics and assumptions that simulate the constraints and challenges for managing the implementation effort in federal agencies.

1.6. Overview

The current praxis is divided into five chapters. The first chapter is the introduction. The second chapter is the literature review and the research contribution to the overall problem. The third chapter discusses how the research was conducted. The fourth chapter discusses the results and activities involved. The last chapter discusses the conclusions and gives recommendations for future research.

2 LITERATURE REVIEW

Different studies in the field of agile software development, as well as federal software acquisition efforts, have provided the foundation for the current study. Additional studies of systems dynamics and their application to software development were also obtained and provided additional background in understanding how agile can be applied to support the current study.

2.1 What is System Dynamics and how can it be utilized?

The Washington University in St. Louis website (2017) defined systems dynamics as the following: "A method for understanding, designing, and managing change. It models the relationships between elements in a system and how these relationships influence the behavior of the system over time". System dynamics is a general-purpose framework that depicts and simulate how dynamic behaviors interact and influence each other as it flows through time. Using a stock and flow diagrams to illustrate the causal relationship between two factors, system dynamics provide an expansive look into how factors influence a particular system activity, business process or other behavioral patterns as exhibited by mechanical, natural or procedural entities (System Dynamics, 2017)

The earliest recognition of system dynamics as methodology for practical applications was in 1956, when Jay Forrester joined the Massachusetts Institute of Technology Sloan School of Management and laid the foundations of system dynamics with his studies on understanding the dynamic behavior of various business-related issues and problems challenging General Electric (GE) at the time (Tang & Vijay, 2001). Predominantly a tool for engineers and scientists in understanding "systems" or a "group

of systems" in its early inception in the 1950s, system dynamics has found more utility for applications outside the scientific and engineering communities to help understand the varied and different relationships not only in the natural world, but also with the processes and activities in daily human society and people's interactions with the world.

One of the earliest applications of system dynamics was in 1972, when Meadows, Meadows, Randers, and Behren III discussed the implications of human growth and the potential underlying the effects that it can impose to the ecosystem. The book *Limits to Growth* showed that as the human population grows and the varying activities that accompany human growth exponentially increase, the ability of the planet to sustain a balanced ecosystem, from temperature to food reserves, habitation, and water consumption, will be compromised if growth is uncontrolled and unrecognized. The authors expanded on Jay Forrester's initial systems dynamics approach and used the science behind system dynamics to underscore the relationships and the impacts of the varying growth elements described, establishing relationships and consequential factors that the authors utilized as a justification and potential resolution for curbing the overall ecosystem imbalance.

In 2013, Ghadge, Dani, Chester, and Kalawsky discussed the utility of systems dynamics and its application to a supply chain risk management framework, showing the varying relationships of elemental risk factors in a given supply chain environment and how to determine these risk relationships and understand the different implications that each factor exerts on every other factor. In this study, the authors determined that the volatility and non-volatility of certain supply chain risk impediments clearly outweighed the other risk factors that were considered (i.e., financial status, transportation limitation,

etc.), as evidenced by the system dynamics software modeling the authors developed in support of their research.

Additional studies from Yamaguchi (2001) and Rosenberg, Riasanow, and Krcmar (2015) also provided information and increased the understanding of not only the functionality of system dynamics, but also its applicability to various environments and how it has a "systems relationship." Both papers utilized system dynamics as a tool to further understand business processes and applications to underscore the versatility and applicability of systems dynamics, not only from a pure esoteric engineering or scientific modeling process, but also in more exoteric business or societal applications that can impact day-to-day activities and societal problems.

Cantwell (2013) further explored the relationship of systems dynamics and project controls and provided a more pointed discussion regarding the use of systems dynamics to better understand how the federal government acquisition process interacts in the traditional cost and schedule performance parameters. Cantwell (2013) utilized existing large-scale federal acquisition programs and examined federal constraints in the development and acquisition process, identifying negative and positive interactions between these federal acquisition project variables and their impact on the overall program development.

In studying these articles, the applicability of systems dynamics as a tool with which to understand the different interactions and impacts of the variables was noted. The articles illustrated that system dynamics, as an approach for studying behavior patterns was a viable methodology for study of process flows and constraint applications. This is a key consideration to the overall premise for this praxis research as it attempts to

understand how federal constraints can impact an agile implementation, and with this understanding, potentially can allow organizations to properly address challenges and limitations in the conduct of activities to support a software development effort within the federal government.

2.2 System Dynamics and Software Development

In an early 1994 article that was presented at the International System Dynamics Conference, Rodrigues discussed the role of systems dynamics in helping understand project controls and the interaction between the variables of project management and prospects for success when managing a given project. Although written nearly 25 years ago, the article still has relevance regarding the overall construct of the current research because of its recognition that although processes may change, fundamental project management considerations, such as schedule and resource management, continue to play significant impacts in managing or addressing challenges in the development of any project.

Rodrigues and Williams (1997) noted the applicability of using system dynamics specifically in software development through the utilization of a generic framework of expected activities, examining how each activity impacts the other activity throughout the development cycle. In the example shown in Figure 2-1, Rodrigues and Williams (1997) defined the various steps in developing a software component for the KDCOM SWB1[5] system and outlined the various relationships and processes involved in developing software. The steps in the model illustrate the path from design to development to the

[5] The KDCOM SWB1 is a Command and Control System for a Destroyer that was developed for the Republic of Korea Navy (Rodrigues and Williams, 1997)

eventual testing and integration efforts, which are the fundamental and generally accepted steps for any software development effort. In a simplified state, the researchers also included high-level management impact variables to be taken into consideration to gain a better understanding of the dynamic behaviors that exist within a software development effort.

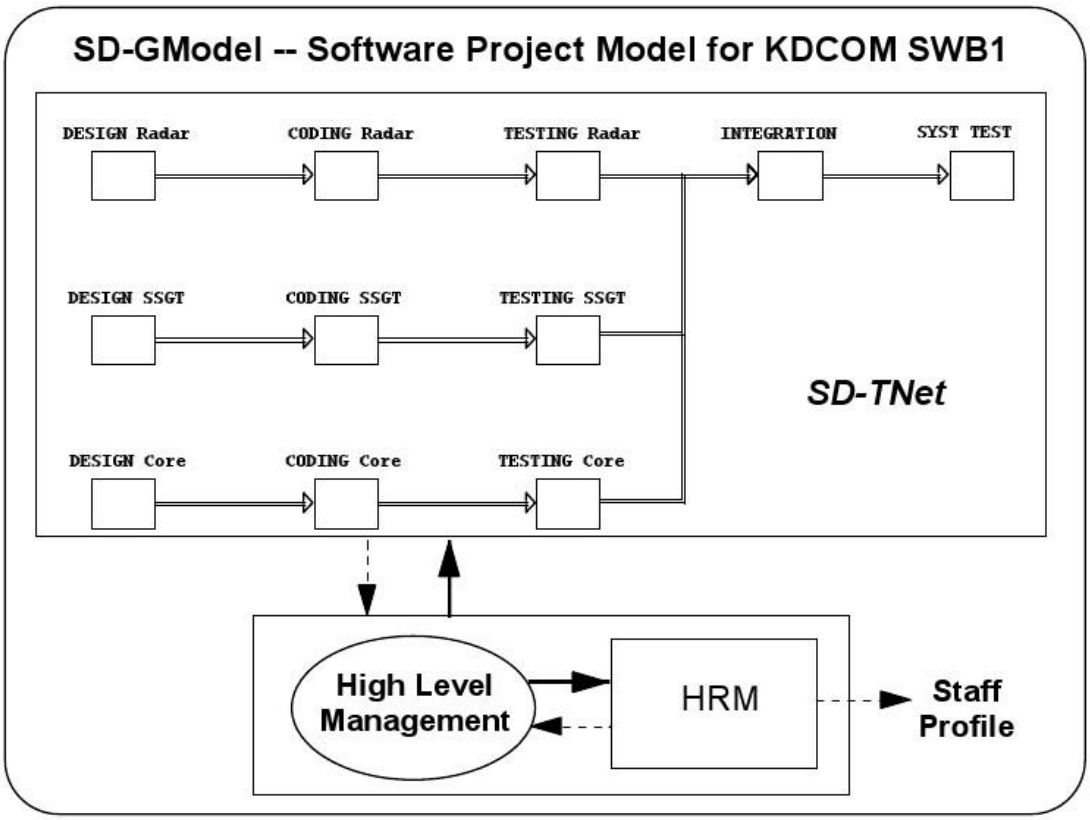

Figure 2-1: Example of software development framework and related activities (Rodrigues & Williams, 1997)

The premise for these articles in defining the activities involved in software development is one of the more pointed focus in the development of this study. For the study, the activities leading up to the actual development effort (in this case an agile effort) was of prime concern as they helped in understanding what the potential pitfalls might be for executing this process. As an example, to implement an agile effort in the

federal government, a contract-type must be selected. Some of the considerations related to this step would be what contract type is amenable to agile (firm-fixed versus cost plus)? What are the limitations and constraints in implementing this contract (flexibility vice cost savings)?

With these "external" factors applied to the software development effort, how do these constraints or considerations affect a software step if proper design or coding is not executed correctly (as in Figure 2-1)? Does this impact the overall cost of the project and if so, is this a consideration for the developer or the federal government (profit versus scheduled delivery)? Understanding these variables and the influence, constraint and impacts these variables exert is the core of this study and serves as the focus for defining how agile can best be managed in the federal environment.

2.3 The Agile Methodology Overview

The TechFAR Handbook (2017) defined agile software development as follows:

A method of software development that is based on iterative and incremental processes and collaboration among a team. It is a methodology for the creative process that anticipates the need for flexibility and applies a level of pragmatism into the delivery of the finished product. The focus is on keeping the code simple, testing often, and delivering functional bits of the application as soon as they are ready.

Figure 2-2 illustrates the iterative process of delivering software when using the agile approach, wherein short development iterations of software development are planned with the explicit purpose of delivering a software product to the customer and in a shorter period. Figure 2-2 describes how the overall software product vision is decomposed into "user stories" and consolidated as the product backlog. The product

backlog is then further decomposed for agile work iterations or "sprints," typically in 1–4 week iterations; within these sprints, the user stories are developed and coded into implementable and shippable portions of the overall product vision. A continuing, cyclic work effort in building software code allows for continuing engagement in refining and identifying the changes or deficiencies in the overall build, thereby allowing flexibility in the software's design and ability to adapt to changing customer specifications (TechFAR Handbook, 2017).

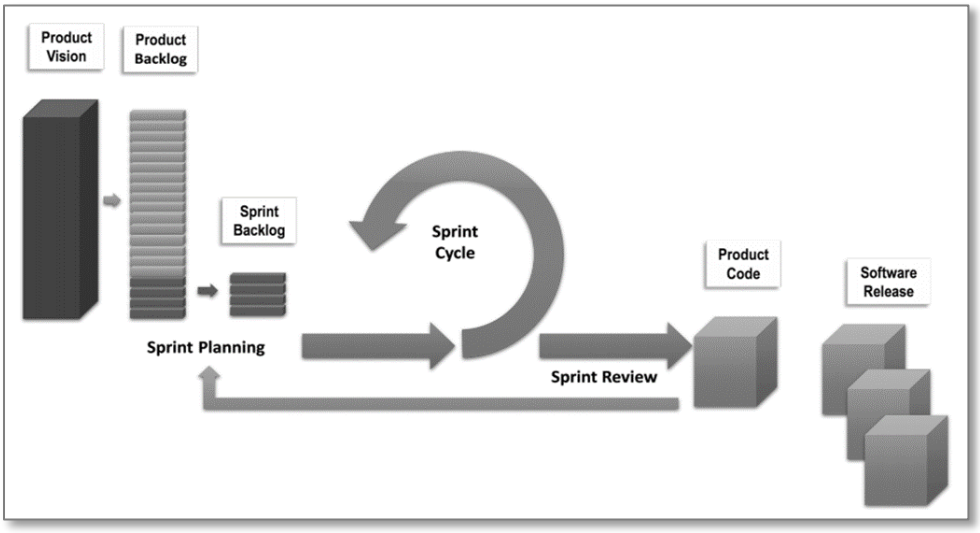

Figure 2-2: The agile methodology of software development (TechFAR Handbook, 2017)

In comparing agile against the traditional software development life cycle (SDLC), Stoica, Mircea, and Ghilic-Micu (2013) stated that the incremental and iterative nature of the agile methodology provides developers with the ability to adapt to changing environments. This advantage helps drive cost-savings in the early identification of software defects, schedule flexibility, and the ability to adopt new user requirements because quicker delivery provides more customer engagement at an earlier stage of the software development cycle.

Abrahamsson, Warsta, Siponen, and Ronkainen (2003) discussed that another advantage in using the agile methodology compared to the traditional methodology for the SLDC is the ability to adjust software development for changing requirements; this is a key advantage that is particularly well-suited for the new technological age of evolving software needs, particularly in the fields of cybersecurity and mobile applications. The rapid pace of change, particularly in these two emerging, critical areas require software solutions that are quick and viable; an agile development can quickly and effectively adjust software development efforts to address the evolving need for the software solution. A lighter and leaner software development effort, designed for software solutions that can be delivered in short time frames, provides an adaptive advantage for customers, and as such, the software delivered is tailored to what the customer truly needs (Abrahamsson, et. al, 2003).

2.4 The United States Federal Government and Agile Methodology

In a 2012 report given to Congress, the Government Accountability Office (GAO) recommended the implementation of a modular software delivery consistent with agile (GAO, 2012). The report reviewed and identified numerous examples of lengthy information technology (IT) projects that incurred cost overruns and schedule delays while contributing little to the overall mission-related outcomes. The report also identified that federal agencies that utilized agile—producing software in small increments with shorter schedules—were more successful in their software development efforts because these agencies were able to deploy software increments in faster and more useful schedules for use (GAO, 2012).

In a follow-up to the 2012 document, the Government Accountability Office released another report to Congress in 2016, indicating a much wider adoption status of agile within the federal government (GAO, 2016). In this report, the GAO indicated that since the 2012 report, federal agencies have begun to embrace the agile methodology, and in the fiscal year 2016, over 64% of federal agencies reported using the agile methodology for their software development and acquisition efforts to date. The report also detailed that for federal agencies reporting the use of agile as their software methodology for development, most of the agencies reported a 6-month planning release cycle, which is well in-line with the agile philosophy of software delivery in shorter periods of time.

The United States Department of Defense (DOD), one of the government's biggest consumers of software has also recognized the role that agile plays in DOD acquisition programs. In 2013, Stephen Welby, the Deputy Assistant Secretary of Defense for Systems Engineering, gave a presentation regarding the adoption of agile within the DOD (2013). According to Welby, as software takes on a bigger role in the development, automation, and acquisition of DOD systems and weapons, the ability to adapt and change to the changing threat environment is through software agility. During the presentation, Welby also discussed changes within the DOD acquisition instructions, known as the Department of Defense Instruction 5000.02 (DODI 5000.02), wherein one of the key revisions for how DOD acquisition acquires software is looking for incremental development for software delivery. For the DOD, Welby discussed that the role agile methodology fulfills in the overall DOD acquisition process is as another viable

tool in achieving the agency's desired end state: to deliver usable software in the fastest and quickest way possible to meet its requirements.

In a 2016 report to Congress' House Ways and Means Subcommittee by the Software Engineering Institute (SEI), Miller, Hayes, and Wrubel discussed the state of agile in the federal government (2016). In this report, the researchers acknowledged that since 2009, the agile methodology has continued to take root in how federal agencies develop their needed software, and the relevancy and use of its benefits are significant. The report continued, saying that for agencies using agile, one of the cornerstones for the implementation of agile in their respective organizations was understanding the limitations of what agile can do in a constrained environment and where the advantages of the agile methodology can be exploited. In their report, the SEI authors presented the following benefits for implementing agile in the federal government as outlined in Table 2-1.

Table 2-1: Benefits of implementing agile in the federal government (Miller, Hayes, and Wrubel, 2016)

Benefits	Notes
Early opportunity for course correction, especially when the environment changes after a program has begun	It is amazing how perspectives change on what a system should do or contain once users actually see the working software. Several times, the SEI agile in government team heard statements such as "Thank goodness we implemented that aspect of the system early. We would never have known to change X if we had just been working with the requirements and design documents."
Early risk reduction, especially in the user-facing areas of the system	Agile's emphasis on early learning by implementing allows a program to focus on risk reduction via implementation rather than speculation through documents that might or might not be correct.
Shorter "idea to realization" cycle that results in fast user feedback for future increments of functionality	In the defense arena, the life cycle is called "concept to capability." The ideas are the same. The faster we get feedback from real users on what we have provided, the faster we can improve the product to meet evolving needs

The SEI researchers also concluded that federal agencies also need to consider the challenges for implementing agile in the federal government. These challenges are noted in Table 2-2.

Table 2-2: Challenges for implementing agile in the federal government (Miller, Hayes, and Wrubel, 2016)

Challenges	Challenge Notes
The need for requirements to be allowed to evolve and the need for ongoing government involvement in their prioritization	Treating initial requirements as final requirements have never led to successful IT projects. [Nielsen 2009] This is primarily due to the volatile technical and operational environments our systems are expected to work within robustly.
Lack of communication and alignment of program goals and strategic initiatives	Although this is a challenge in any IT Transformation setting, the effect of poor communication and alignment is more visible in settings where the delivery cadence is on the order of weeks rather than months. Frequent deliveries allow the course correction that may be needed to mitigate this issue.
Lack of workforce knowledge, skills, and abilities related to overseeing agile projects	This has been a consistent challenge across multiple programs adopting agile and lean methods. Early adopters struggle to find training to prepare them for new duties. Invariably the second wave of adopters gets more training than the first, after the experience of the first wave in adjusting to government oversight obligations.
Difficulties in aligning acquisition policy with small-batch approaches such as agile	Slowly the acquisition policy environment is changing; until recently, enabling incremental delivery, review, and production of documentation and software products was challenging. We have seen multiple programs which had to deal with the overhead of large-scale document reviews at the same time they were trying to implement small batches of the software described in the documents.

2.5 Challenges of Implementing Agile in the Federal Government

In a 2010 dissertation paper, Cho discussed identifying the issues and challenges in implementing an agile software development effort. Her research identified four key elements that are significant contributors in implementing agile; Cho recommended that agile practitioners take these issues into account as part of a formal process when

planning and defining strategic activities for an agile development effort. The four key elements are as follows:

1. Human resources management

2. Structured development process

3. Environment

4. Information systems technology

Cho further stated that these four elements invariably play key roles in influencing and affecting how the other elements behave, and each element can serve as an enabler or disabler in the success of an agile implementation effort (Cho, 2010).

Similarly, in 2015, Brown examined the relative impacts of 12 implementation factors for agile projects and assessed the significance of each factor in the overall success or failure of an agile project. Using a regression analysis and determining the relative impacts of each weighted variable, Brown concluded that four of the 12 factors, namely project type, project schedule, project nature, and delivery strategy, were significant enough to be considered critical success factors for agile project implementation and success (Brown, 2015). However, for both Cho's and Brown's studies, the sampling environment for agile implementation was in commercial or private-sector industries.

Mueller (2014) conducted a study using the Delphi method and identified the factors that influenced agile methodology implementation in federal agencies. Using a sample study of agile practitioners in the federal government, he initially determined

there are 21 factors that are impactful in how the agile methodology is implemented in an agency. He also conducted a more pointed sampling determination, isolating responses by gender, position, and other categories. Mueller concluded that the top five factors that contribute to a successful agile implementation in federal agencies are as follows: organization culture, leadership involvement, user involvement in the development cycle, change management, and overall team environment. This research provided a more pointed discussion and directly supported the premise of understanding and addressing the problems in implementing agile in federal agencies by narrowing down factors that can impact an agile effort.

Chang and Modigliani (2015) conducted a study regarding the challenges of implementing the agile methodology in federal agencies, and the authors presented their findings at the 12th Annual Acquisition Research Symposium. The presenters noted that as the adoption of the agile methodology continues to increase in the federal government, the fundamental behavior and understanding of how to tailor existing policies and procedures impact the ability of agencies to successfully implement agile. Citing DOD process and procedures, the presenters underscored the lack of knowledge to tailor DODI 5000.02 to incorporate agile development methods in major software systems acquisitions, thereby creating barriers in allowing agile as a means to develop software, which was shown to be a causal factor for DOD acquisition programs not fully adopting the agile methodology. In concluding the presentation, Chang and Modigliani discussed the three key elements that government program managers need to focus on: the agile requirements process, contracting management for agile, and agile program structuring. These fundamental elements lay the foundation for a successful agile implementation

(2015). Chang and Modigliani's research and the SEI (2016) report to the House and Ways Means subcommittee serve as the primary backdrop for the overall intent of understanding how agile can be successfully implemented by the federal government.

While these studies pointed out the challenges and factors that can or will influence an agile implementation effort, it was noted that there was not enough discussion or analysis conducted to truly understand how these factors impact the implementation process, why factors become critical to an overall effort and what can be done to plan or understand implementing agile within the constraints of the federal environment.

2.6 Requirements and how it relates to agile in the Federal Government

Fundamentally, requirements, as they apply to federal government acquisitions and development, are the primary driving factors for any task to be undertaken. For the agile methodology, requirements define what kind of software is envisioned and how it will be broken down to meet the user's needs. The TechFAR Handbook (2017) identified requirements as the driving element for what the desired product vision entails, and this vision feeds the product backlog, an item which contains the prioritized work effort for the development team. The TechFAR Handbook discussed that development iterations draw from the product backlog for work to be accomplished during these time blocks of the development and completion of these user stories, are grouped together as a "software release" (2017). Logically, proper requirements definition implies a good backlog development; this however, has been one of the more complex challenges for any government acquisition or development effort as defining requirements in the federal environment is one of the more challenging activities to conduct. In having to answer to

multiple stakeholders and project interests, requirements are usually set at a high level definition wherein interpretation and assumptions are usually accepted as "desired capabilities". Defining requirements becomes a convoluted effort once the actual development process begins as changes continue to occur with requirements, necessitating a continuing change in the backlog. Arguably, this is where agile is supposed to be applicable; the problem however, is when the constraints of the federal environments are applied, such as contract type (funding type), schedules (time to execute), or performance (incorporate information assurance requirements), the ability for agile to address the changes are nullified.

This same premise argument was also presented by a study from the Software Engineering Institute (SEI) (2016), where the need for requirements management and the ability of federal program and project managers to manage changing requirements is a primary factor in ensuring the implementation of agile efforts. Huckabee (2015) discussed the need to implement a requirement engineering process in support of agile development efforts within federal agencies. He pointed out that defining requirements along "use-case" capabilities of deliverable software solutions is a critical enabler in ensuring that agile processes remain in place. Huckabee noted that because agile processes revolve around the development of software in support of the product backlog and because the agile development cycle promotes an iterative cycle of deliverable work within a given sprint timeline, the software's definition of requirements should address the desired end-state capability of what the user needs. Relating requirements to an "actionable" deliverable, that is, the software delivered can perform x calculations, can help validate the completeness for said requirements and define the product build.

Invariably, defining the requirements along the lines of what can be delivered and can be accomplished in each sprint time frame also enables continuing collaboration between developer and customer (Huckabee, 2015).

Overall, the intrinsic relationship of requirements management and agile development are two of the most fundamental linkage for understanding how agile can be implemented properly in the federal government. Knowing that requirements will evolve and understanding that software is a malleable product, the ability to relate limitations (such as contracts, budget, etc.) and trade-offs (what capabilities can fit in the development effort) in what needs to accomplished is an essential management and strategic consideration for federal agencies implementing agile.

2.6.1 Role of information assurance in agile development

The rapid growth and adoption of agile in developing software for the federal government has also created a unique problem in dealing with the need to address the requirements of software security and information assurance. In a paper written by the Software Engineering Institute at Carnegie Mellon University for the Office of the Secretary of Defense, Bellomo and Woody discussed the implications of information assurance (IA) requirements on software delivery (2012). In an interview with agile program managers, the authors described how the current IA process for accreditation was described as “tedious, cumbersome and not conducive to agile development” and how it completely negated the potential advantages of early or efficient delivery of software to customers. The authors also discussed the risks involved and how agile development backlogs are impacted because of the late involvement of IA stakeholders in the development process, negatively affecting the advantages gained by using agile. The

authors further discussed the additional impacts on the overall agile implementation construct and development effort of IA, including testing for deficiencies, collaboration deficiencies, and the overall non-cohesive implications of actions and development activities to support IA requirements. In conclusion, Bellomo and Woody (2012) stated that IA activities play a significant role in the overall success and challenges faced by federal agencies in implementing an agile methodology.

The NIST Special Publication 800-53, *Security and Privacy Controls for Federal Information Systems and Organizations* (2013), provided a good, initial baseline for understanding and defining information assurance requirements for software development. The document aimed to provide a systematic approach for defining security controls and processes and how they relate to information technology products and systems, and it gave a baseline requirement for IA. This baseline requirement is an important aspect for any software development effort as it helps define the scope and the backlog for the overall effort from the beginning of the project vice integration at a later stage. Integrating these IA requirements early in the project development cycle minimizes disruption or late-stage additional work that can potentially derail the projects schedule and cost parameters.

In reviewing these two documents, the impact of IA on whether it is included in an agile backlog or as a post-agile development activity cannot be underestimated. Software development in federal agencies continues to struggle in trying to address and incorporate IA requirements due to its ever-evolving posture or because the bureaucratic processes negate agile advantages. In either case, understanding or having an

appreciation for potential impacts to software delivery due to IA requirements should be at the forefront of any agile implementation strategy for federal agencies.

2.7 Contracts Management and Agile in the Federal Government

The Federal Acquisition Regulation (FAR) is the primary federal document that governs all acquisition activities for United States federal agencies. The document, comprised of 34 parts, "is a codification and publication of uniform policies and procedures for acquisition by all federal agencies" and is the primary source of information for all activities in the FAR system (FAR, 2017). Although FAR is the overarching directive for U.S. federal agencies, individual policies, guidelines, and directives exist within each agency, and in general, these agency-specific directives, policy, and guidelines are more restrictive and more "tailored" to the agency's specific needs (FAR, 2017). Additional resources that government federal acquisition activities could include are the code of federal regulations (CFR) for the codification and publishing of agency regulations and policies (U.S. Government Publishing Office, 2017) and the DoD's federal acquisition regulation supplement (DFARS) for DoD-specific regulatory provisions, particularly in relation to foreign military sales and the North Atlantic Treaty Organization cooperative projects funding and contract execution (DFARS, 2016).

Within FAR's guidelines, Information Technology (IT) Reform Action 15: Issue Contracting Guidelines and Templates to Support Modular Development, 25 Point Implementation Plan to Reform Federal Information Technology Management (2012) recommended in Table 2-3 that contracts should be in support of incremental or modular software development efforts. Most agencies however, struggle to understand how

contract type can influence the agile tempo as the familiarity with contract-type and tie-ins to software development is not well understood. Some of the challenges include awarding of firm-fixed priced contracts for evolving software requirements for research and development projects, cost-plus contracts that continue to grow in scope and incomplete software development due to time-and-material contract awards.

In 2003, Alleman, Henderson, and Seggelke discussed how the award of government contracts, such as firm-fixed price, cost-plus and time-and-material contracts, impact agile development in federal agencies as it relates to the type of work in needs to accomplish. In their research, the authors noted that contracts help define the type of work effort that can be expected from agile contractors, wherein risks are defined depending on the contract type that the government uses. The authors stated that for firm-fixed contracts, where the risks are higher for the developer, agile development is more straightforward and has very little leeway for adjustments in work effort without incurring additional costs. For cost-plus contract types, where the risks are higher for the government, contractors are more amenable to adjust their work effort, whether it's because of a requirements change or additional scope, because the ability of the government to provide additional funds can change (Alleman et al., 2003). Alleman et al. (2003) concluded that for agile development, selection of the right contracting vehicle is an essential part of implementing agile in federal agencies and validates recent findings from SEI (2016) on what factors are significant in implementing agile in a federal workspace.

Table 2-3: Contracts summary for implementing agile in the federal government (TechFAR, n.d.)

Type of Contract	Recommendation
Fixed-price contracts and task orders	In fixed-price contracts and task orders, the price is negotiated before award and payment are tied to the delivery of a completed product or services. Fixed-price arrangements should be used for the acquisition of COTS and development services for software projects or increments where the Government knows the specific functional characteristics that will satisfy its objectives. In these situations, fixed-price contracts provide the greatest incentive for the contractor to control costs and perform effectively.
Cost-reimbursement contracts	Cost-reimbursement contracts are appropriate when uncertainty in the requirement and the risk of failure cannot be managed by the contractor within economically reasonable bounds. This uncertainty may be due to lack of knowledge regarding the effort needed to meet a defined requirement, including the Government's and industry's lack of cost experience in performing the work. Alternatively, the uncertainty may be due more fundamentally to a lack of knowledge about what is possible and practical, as is often the case with research or leading-edge innovation.
Time-and-material and labor-hour contracts	Time-and-materials (T&M) and labor-hour (LH) contracts may help agencies accomplish tasks for a reasonable cost where the needed amount of labor effort cannot be specified in advance, such as when an IT office must diagnose the cause of a system failure or an agency is reacting to an emergency. Before an agency uses a T&M or LH contract, or places a T&M or LH order (e.g., including under GSA's Federal Supply Schedules), it must make a determination that it is not possible at the time of placing the contract or order to accurately estimate the extent or duration of the work to anticipate costs with any reasonable degree of certainty and no other authorized contract type is suitable. Because a T&M or LH contract provides no positive profit incentive to the contractor for cost control or labor efficiency, appropriate Government surveillance of contractor performance is required to give reasonable assurance that efficient methods and effective cost controls are being used.

2.8 Role of Schedule Management and Agile in the Federal Government

Bird (2010) discussed in his paper how schedule constraints, as imposed by the customer, is a significant driver for any agile software development effort. Bird (2010) concluded that software development using agile clearly demonstrated a significant

advantage in schedule delivery for working code to the customer because of its rapid and continuing focus on realistic and deliverable tasks, instead of trying to provide an all-encompassing software solution that traditional SLDC espouses. Bird pointed out that schedule advantages, as a function of capability, also need to be treated carefully because the prospect of deviation, whether through changing requirements or because of overall software build demands, can change the overall work effort compared to the original purpose of the development (2010).

Ronkainen and Abrahamsson (2003) discussed how agile methods are impacted by other constraints in the development of IT systems. Using hardware and schedule demands as constraints in development, the authors noted agile's ability to accommodate varying development requirements for the overall system gave the developers an advantage and validated the point that agile is more adaptive in meeting consumer software needs. Furthermore, according to the authors, because of its ability to adjust to changing design, the software developed using the agile methodology provides a more robust and integrated software support for emerging components and is clearly more efficient than the traditional SLDC methodology. Ronkainen and Abrahamsson (2003) concluded that agile, if properly managed, has clear advantages in not only meeting the overall schedule demands, but also in allowing for a more cohesive and integrated work effort in developing systems that are hardware and software dependent.

Schedule management, as a driving constraint for agile, is a significant factor to consider for implementing agile in the federal environment. For the developers and the federal government, this factor defines what can be accomplished in a set amount of time with a determined amount of cost. From a project management perspective, knowing and

understanding the relationship between schedule and expected performance for the software to be developed, is a fundamental skill in order to take advantage of potential trade space in capability development or defining acceptable products if not all capabilities were accomplished at the end of the scheduled timeframe.

2.9 Performance Management and Agile in the Federal Government

Cao, Ramesh, and Abdel-Hamid (2010) discussed the impact of an agile team's performance and how it impacts the overall agile methodology. The authors noted the commonly used approach of paired programming in agile teams and how it impacted quality and refactoring efforts. Paired programming, wherein a developer is teamed with another developer in creating user stories, allows for a more efficient sharing of work effort. The authors highlighted that paired programming showed to have a more advantageous capacity in how the agile team addresses the development effort, including the quality rework and refactoring, because of gained efficiencies gained from team performance versus from a strictly individual effort. The authors also pointed out that the use of paired programming also gave the overall agile development effort the ability to check and cross-check in a more timely and efficient manner because developers could be more diligent in refactoring their codes due to shared effort and multiple common looks at the code. For the purposes of this study, the scoped was limited to understanding the impact of paired-programming teams in order to streamline the activities that need to be conducted in evaluating effects to the overall agile development. As there are other techniques in understanding or enhancing performance effectiveness for agile teams, such as non-paired teaming for development, intent was to pattern the study with the most common approach of team development.

With the increasing adoption of the agile methodology for developing software, Lalsing, Kishnah, and Pudaruth (2012) studied the impact of "people factors" and in the overall construct of agile team development. Because agile development focuses on adaptive and customer-relationship teams, the authors discussed that team performance and relationship within agile teams are significant drivers in implementing success for any agile development effort. The authors also defined agile team sizes and how a team's size could impact the efficiency and rate of delivery. Lalsing et al. (2012) emphasized the need not to underestimate team composition and cooperation because it relates to the overall agile development work effort. Understanding the factors that go into analyzing how a team might perform is critical in minimizing the challenges and barriers in utilizing agile in support of federal government software development efforts

2.10 Contribution to Research

Although many publications and research exist in understanding the agile methodology and the advantages it offers, there has been very little research on the effectiveness of implementing the agile methodology in the federal government. In the commercial world, the use of agile software development has been firmly established as a means of developing software for consumer applications. Rapid consumer demands for new applications and software solutions dictate that developers utilize agile in order to deliver software products in the quickest and most expeditious manner in order to meet the demand. With minimal constraints imposed, commercial development of software using agile has quickly transformed how industry delivers applications, software solutions and various other software components that clearly takes advantage of quick,

rapid and successive iterations of software in order to satisfy the continuing changes in demand and functionality.

While most federal agencies are mandated to provide status of the IT investments within their respective agencies for a high-level government spending summary, true reporting of their success and failures and usable lessons learned within these agencies regarding the implementation of agile are anecdotal at best, with very little relevant information on effective strategies or recommended courses of action. For the federal government, the promise of this development and delivery approach providing software solutions at a rapid pace is very much an attractive prospect given that software solutions play key roles for the federal government, including for the government's business operations, cybersecurity, database management, and more.

Consequently, although the agile methodology is becoming the preferred process for software development and acquisition in federal agencies, it is important to note that legally, the federal government must adhere to defined processes and procedures to acquire goods or services. For the federal government, the primary binding document to acquire or obtain services is the contract. This document serves as the primary mechanism for funding the allocation and budget and delineating the types of products or services that the government requires. The contract defines what kind of product or services the government requires (requirements), when these items are needed (schedule), and how the government will pay for the goods or services (contract type). As such, the contract defines the very basic premise of the constraints for any agile implementation because it serves as the fundamental document for how the agile process needs to be

implemented to support the binding agreement and provide goods or services to the government and the software contractor.

Within the construct of the agile methodology itself, various constraints also exist that impact how the process can effectively deliver the software product. As with any software development effort, the varying impacts of quality (added work in backlog), rework (added work and schedule to scope), and resource limitations (less capability to conduct work effort) will have a direct impact on the effectiveness of any methodology or process development. In all, a simplified model of how the agile methodology is implemented in the federal government is shown in Figure 2-3. In using this model, tangible metrics and values can be applied for determining behavior as the development efforts progress through time. For federal program or project managers in charge of managing an agile software development, these elements, or "constraints," that exist within the contract and the agile methodology itself also serve as implementation monitors.

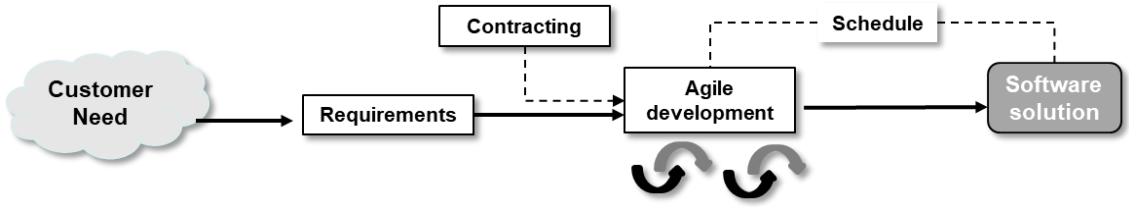

Figure 2-3: Implementation model for the agile methodology in delivering software

3 RESEARCH METHODOLOGY

3.1 Goals and Objectives

The main goal of the current research is to shed light on the dynamic consequences of federal acquisition constraints for the requirements, cost, schedule, and contract type on agile software projects. With the continuing shift of federal investments and software acquisitions to an agile environment, the need to understand how these constraints impact agile software development takes on a more distinct importance; if understood better, the government could fully utilize the advantages that the agile methodology can provide, allowing project and program managers and decision makers in federal organizations know how to resource and manage efforts using the agile methodology.

Ultimately, understanding these constraints and dynamic interactions will also provide more effective insights for software developers who want to use the agile methodology when providing software solutions to the federal government. For developers and agile practitioners, knowing and understanding the varying impacts of these constraints can help shape how they resource and manage changes in the development effort and provide insight in determining effective cost-management actions or cost implications for actions or activities that federal managers may impose on agile implementation efforts.

3.2 Scope and Limitations

The main focus of this study was for agile projects with distinct software delivery requirements. It does not include or evaluate additional software activities for software integration or enterprise-wide full software solutions that have a multilayer contract,

development methodology, or funding cycles (sample projects considered were smaller parts that fed a bigger solution or effort). This study also utilized open-source data publicly available through the ITDashboard.gov website to minimize potential complications with security or additional information assurance requirements that could skew the comparison of the results (OMB, 2017).

Ideally, a deeper discussion of both software development performance and information assurance impacts as they relate to federal agencies would also add additional value for understanding both the costs and schedule implications. This, however, was deemed too detailed and encompassed a much bigger work effort that could not be accommodated in the time-scope requirements for the current study.

3.3 Research Method

The present study utilized a simulation research method to understand the dynamic interactions and consequences, such as performance degradation, cost control, scope creep, etc., among the selected parameters for agile implementation and activities. The current study followed a standard systems dynamics framework, as discussed by Martinez and Richardson (2001), as shown in Figure 3-1. Six activities—(1) problem identification (2) system conceptualization, (3) model formulation, (4) model testing and evaluation, (5) model use, implementation, and dissemination, and (6) design of learning strategy and infrastructure—define the overall organization of the framework used for the current study. Minor changes in the framework, particularly for design of learning strategy and infrastructure, were conducted, wherein, the conclusions of the study were utilized more as a possible framework for additional studies versus reintegration into the learning process of the system dynamics model.

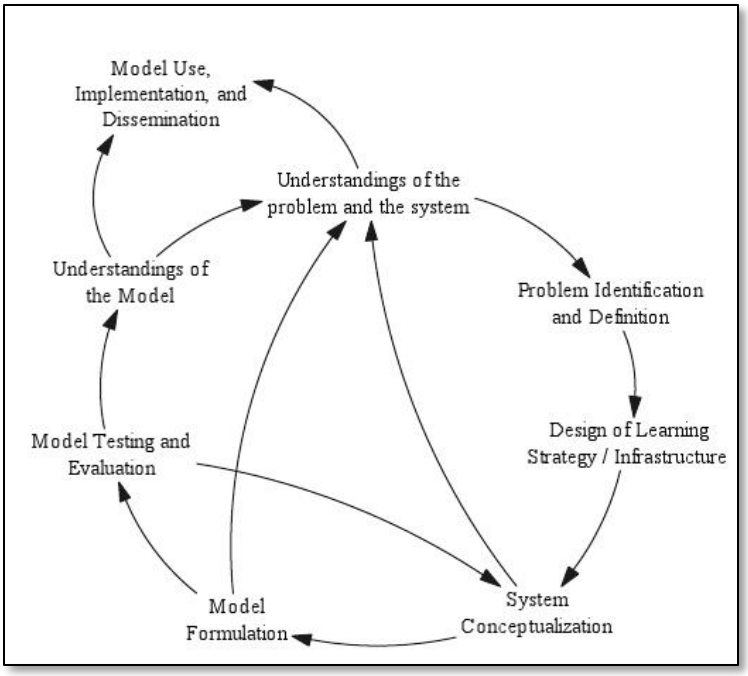

Figure 3-1: Overview of the systems dynamics modeling approach (Martinez & Richardson, 2001)

This current study also utilized Glaiel's (2012) agile project dynamics (APD) model to identify the various agile components for study considerations. Figure 3-2 is an illustration of these agile components as they relate to the traditional project management constructs of cost, schedule, and performance parameters. These selected components were then incorporated as baseline considerations for agile software development and high-level interactions that can potentially impact overall agile development efforts. Although the model provided a good foundation for some of the causal factors to consider, the model was geared primarily for a commercial application of the agile methodology; hence, additional factors (contract type and information assurance parameters) were also incorporated into the baseline model to address specific agile considerations for federal agencies.

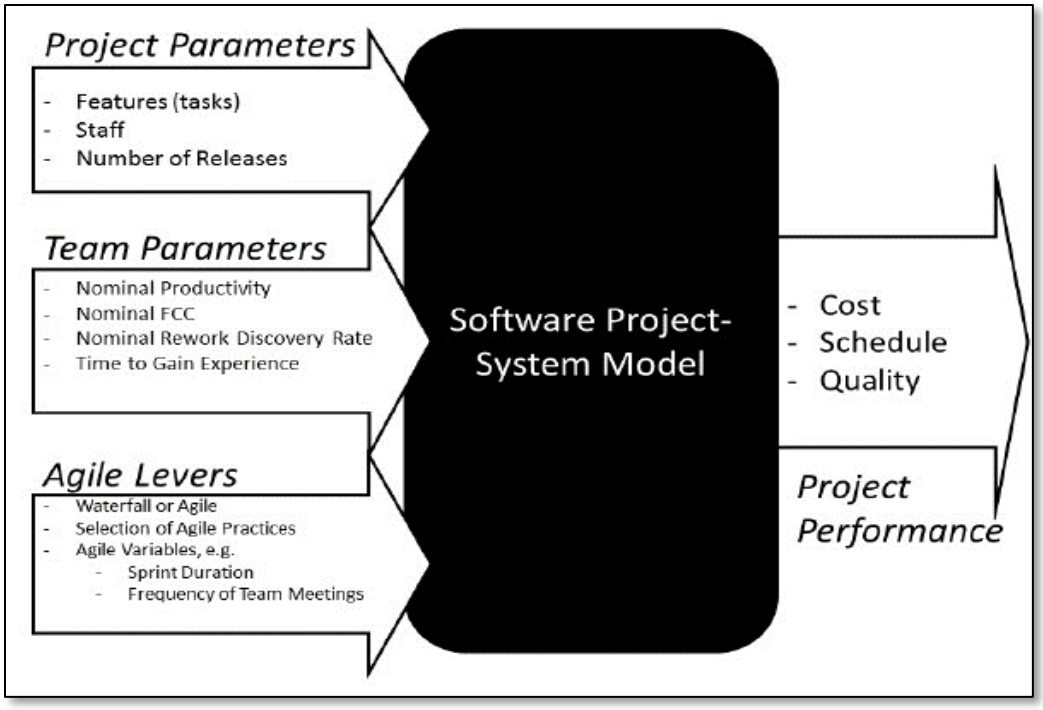

Figure 3-2: The agile project dynamics model (APD) (Glaiel, 2012)

3.4 Base Model Overview

The baseline model for this current study is shown in Figure 3-3. To be easily understood and potentially used in future studies, the model design was simplified as much as possible.

The focus for the model is the interaction of the applied constraints to the agile processes and determination of the impact of each element's sub-variable to the desired end-state information for use in decision making. As such, the understanding that the parameters utilized were not primarily reviewed for their final end-state value but rather for their behavior and predicted behavior, as these variables interact throughout a given time period, was of more intrinsic value and served as the basis for the discussion and analysis when using the base model. In using this approach, the limited available data did not prevent the current study from providing valuable interaction trends and results,

hence allowing the researcher to provide relatively conclusive evidence on how agile implementation activities are impacted by constraints in the federal acquisition processes.

The Vensim simulation software was also utilized for this study, and the baseline model served as the primary vehicle for variable inputs and design changes. Being a system dynamics simulation model, the software also provided valuable insights into the comparative analysis of the variable inputs and gave time-based behavioral changes for the changing parameters of the model. This proved extremely useful, particularly in the analysis of the information assurance variables as it was applied to different timelines during the software development cycle in order to understand how it impacted the overall effort.

The use of publicly accessible data was also paramount for the development of the model because it ensured that sensitive or potentially sensitive software acquisition or development efforts and information were not compromised. Using information from the federal government reporting website, ITDashboard.gov, various information technology investments and the corresponding results were gathered, analyzed, and tested (OMB, 2017). This information provided a baseline understanding of how agile has been implemented within the various federal government agencies, the contract structure for the software acquisition, methodologies used, schedules, the intent of the software as it relates to overall agency needs, and various other information that assisted in developing the parameters for the current study.

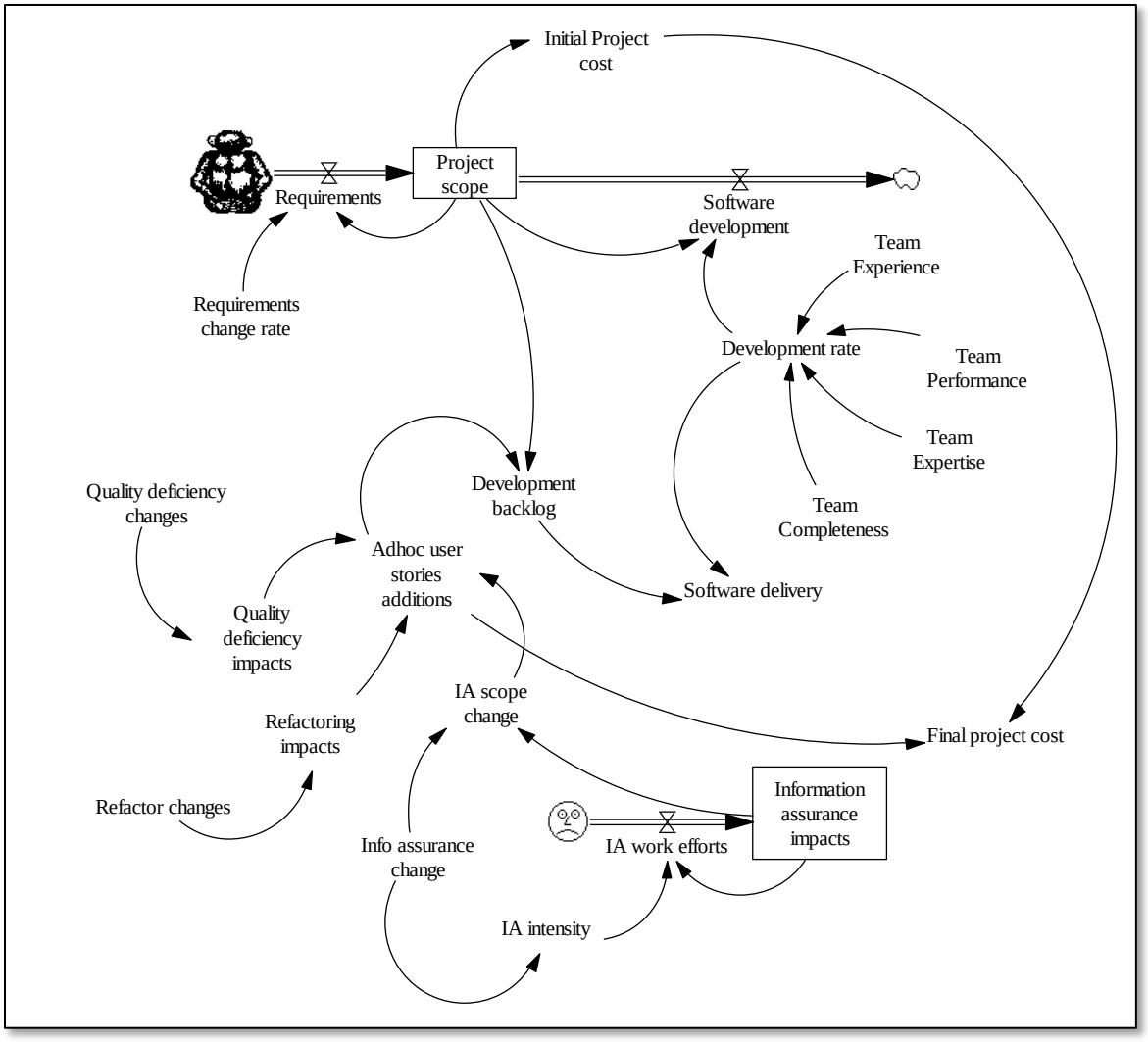

Figure 3-3: System dynamics baseline model for agile implementation

In all, the model highlights key aspects that are particularly applicable to all federal agencies and that are fundamentally relatable to the activities conducted in any agile development effort. Specific tailoring can be conducted for additional agency-specific activity as to how agile can be implemented, but overall, the factors and variables utilized are similar across all federal agencies.

3.4.1 Defining the baseline

As previously discussed, data for the model were obtained primarily from the ITDashboard.gov website. The website is a federally funded portal that was initially established in 2009 to provide the public with transparent information regarding major IT investments within United States federal agencies. Since its inception, the website has expanded its ability to capture publicly accessible data regarding not only federal investments, but also the state and status of ongoing major IT activities that are being conducted by the federal government (sensitive or classified activities are not included in this category; Vanroekel, 2012).

To scope the data used in this model, published information from 2013–2016 for IT investments from the website was obtained. Of the 4,514 reports initially captured, 280 completed projects that utilized agile were identified after weeding down the projects to ones that are less than $20 million and had an approximate development period of 2 years (OMB, 2017). Constraining the period of performance to 2 years and cost to less than $20 million were done for the following two main reasons:

1) Fundamentally, agile projects are not designed for long-term software development efforts (beyond 2 years) and are the most effective for delivering software in a short time span.

2) Federal contracting above $20 million can potentially require earned value management reporting that not only can skew the software development effort by adding additional activities not related to software development, but also can also complicate the overall implementation process with the additional work.

Of the remaining 280 projects, another filter was applied for the availability of data (contracts, business cases, requirements use, etc.), and the resulting data for the model were as follows:

1) Forty-six agile completed projects from six federal agencies (for 2013–2016)

2) Period of performance: < 2 years; Cost: < $20 million

3) Cost average: $5.6M

4) Schedule average: 14.5 months

Cost and schedule averages were the average values for the selected projects; although not specifically utilized in building the model, contract types for the different projects were also taken into account for the evaluation and eventual discussion points for the research.

3.4.2 Factors for the baseline

In defining the parameters that were considered for the baseline model, both "hard" and "soft" factors were considered. "Hard" factors are variables that have defined values that are intrinsic to the implementation process. FAR defined contracts as "mutually binding legal relationship obligating the seller to furnish the supplies or services (including construction) and the buyer to pay for them" (FAR, 2017). For such agreements to occur, three key essential elements must be considered for contracts, as follows:

1. Requirements: What is being procured, or what is being acquired?
2. Schedule: When will these products be delivered?

3. Cost: How much is the agreement going to cost?

These three factors are defined as “hard” factors because they pertain to an end-state product being delivered, amount of money spent, and the timelines or dates that must be adhered to between the government and the contractor as a business relationship between the two parties.

“Soft” factors, on the other hand, were considered for the baseline model; these variables are still influential in how the implementation process is conducted, but the values are less defined, and assumptions can or were taken into consideration to validate the behavioral aspects of the implementation process. Identifying these variables took several iterations, and reviews of previous agile implementation models were conducted; invariably, two distinct studies outlined the final variables to be considered. Lalsin, Kishna, and Pudaruth (2012) discussed the impact of people as it relates to agile development and the overall process. According to the authors, the ability of the team to perform, understand, and execute agile software development is a function of a common understanding of the roles and responsibilities necessary to execute an agile effort. This approach was further validated by Cao et al. (2010), who further proposed defining the critical factors for consideration in agile production, as follows:

1. Human resources management
2. Refactoring and quality of design
3. Change management
4. Agile planning and control
5. Customer involvement

As Figure 3-4 illustrates, the varying interactions of these factors, as proposed by Cao et al. (2010), form a distinct influence on how agile is implemented and impacted throughout the development effort.

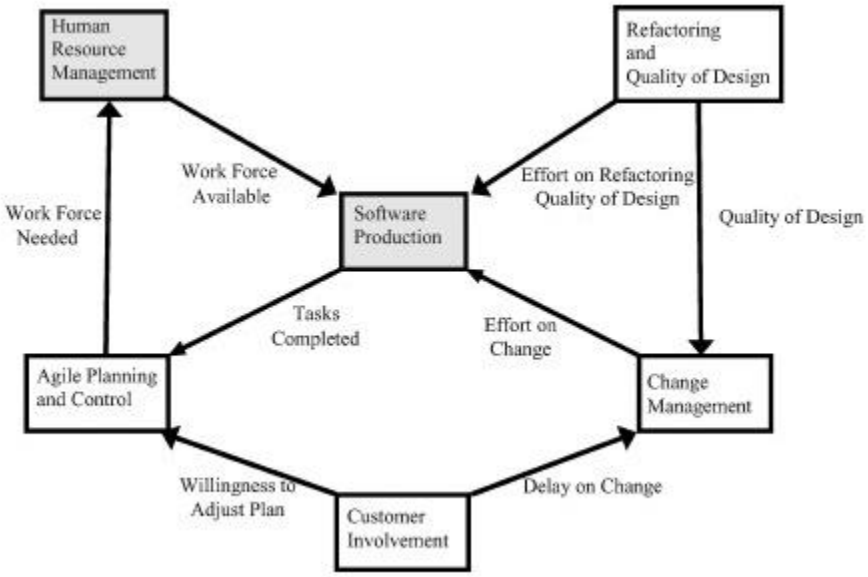

Figure 3-4: Interaction of critical factors for agile implementation (Cao et al., 2010)

Using these studies and others that were reviewed for relevance, a summary of the variables utilized for the model is discussed in Table 3-1; here, the justifications and rationale for consideration for inclusion in the overall study are provided. It is important to note, however, that the overall scope of the current study is to understand the varying impacts of the constraints when they are applied to the process of agile implementation within a federal environment. As such, although certain assumptions were analyzed (such as the development rate and impact percentages), the goal was to understand the varying impact as it relates to certain parameters.

Table 3-1: Factors selected for baseline model

Factor	Rationale for consideration	Methodology applied	Metrics used
Requirements	A key element in any federal contract or business transaction. This defines what the product or service being acquired is.	An agile project is sized according to the product desired. For agile development, this determines the number of user stories that will need to be addressed.	A number of user stories based on the initial development rate and timeline; change in requirements (max 20%)
Schedule	A key element in any federal contracting or business transaction that can also be known as period of performance or defined delivery schedule	Agreed upon date for the delivery of a final product between the government and contractor; expressed in terms of days or months	Utilized days (D) or months (M)
Cost	Overall cost for the product or service that is mutually agreed between the developer and government for work or product provided	Mutual agreed upon cost for the work effort as measured in dollars	Utilized $K for thousands and $M for millions
Development rate (velocity)	Rate of development for developing software, as defined by the agile methodology	Rate of development, as defined by agile effort	Assumed to be an initial capacity of 200 user stories divided by 2-week sprint cycle
Quality deficiency	Quality is the amount of rework conducted for a faulty or nonworking software build.	Amount of additional work based on quality issues	Percentage of original work (max of 10%)
Refactor impacts	Refactor is the amount of additional work needed to improve code functionality without altering the overall code.	Amount of additional work	Percentage of original work (max of 10%)
Team performance	Ability of the team to perform as planned	1 of 4 contributors to overall development rate	Default 25% impact
Team expertise	Ability of the team to know what needs to be executed based on individual knowledge	2 of 4 contributors to overall development rate	Default 25% impact
Team completeness	The overall team composition that is adequately resourced according to the scope of work	3 of 4 contributors to overall development rate	Default 25% impact
Team experience	Overall team knowledge as it relates to familiarity of work to be conducted and knowledge of agile requirements	4 of 4 contributors to overall development rate	Default 25% impact
Information assurance impacts	Ad-hoc additions to existing work due to federal requirements for security and information assurance	Additional work and timeline consideration for work	Additional work of 2% and 5% of user stories in varying schedule applications

3.4.3 System dynamics application

In 2012, Choopojcharoen and Magzari discussed the mathematical principles behind the system dynamics methodology; they noted various models and processes where systems dynamics have been applied as a modeling tool and the mathematics involved with each model. Using this study as a guide, a basic mathematical discussion is presented as the basis for the system dynamics model used for the current study. The basic premise of the model and the utilization of the system dynamics approach in understanding the cascading effects of constraints and variables in the implementation process for the agile methodology is defined in Figure 3-5. The diagram illustrates the basic inflow and outflow relationship between the desired software product as identified by a federal agency and the ability of the software developer to develop the product using the agile methodology. Requirements, in this illustration, form the basis for the project scope and establish the amount of work that will be undertaken by the developer. Time is a function of the developer's ability to complete the task based on the developer's capacity to do work using the agile methodology.

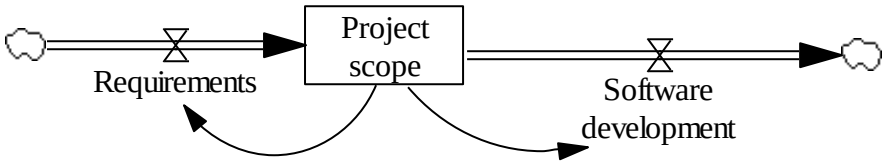

Figure 3-5: System dynamics representation for study

This capability can also be represented as a function of the following:

Inflow-outflow = rate of change of the project scope = dx/dt *(Equation 3.1)*

In this example, if the requirements do not change during the development time frame, the project scope will be accomplished at the predefined time based on the agile development rate of developing software.

Using this as the foundation for continuing the systems dynamics approach, it can be assumed that the factors that impact the project scope, such as requirements changes based on customer inputs, errors, or technical deficiencies, can and will impact the size of the project scope. The same rationale can also be applied to the software development capability when it comes to internal and external factors affecting the ability to complete the task; these factors can include degradation in overall team ability, higher rate of change for work to be accomplished, and timing of work that needs to be accomplished to meet the overall delivery schedule.

In essence, for this system dynamics approach, the overall mathematical representation for the model is as follows:

$$\left[\sum \frac{\Delta x}{\Delta t} * \left(\frac{\Delta r}{\Delta t} + \frac{\Delta q}{\Delta t} + \frac{\Delta rf}{\Delta t} + \left(\frac{\Delta ia}{\Delta t} * t\right)\right) - \sum \frac{\Delta y}{\Delta t} * \left(\frac{\Delta ex}{\Delta t} + \frac{\Delta c}{\Delta t} + \frac{\Delta ep}{\Delta t} + \frac{\Delta p}{\Delta t}\right)\right] = T \quad \textit{(Equation 3.2)}$$

where:

$\frac{\Delta x}{\Delta t}$ = the rate of change for the inflow of work;

$\frac{\Delta y}{\Delta t}$ = the outflow capacity to do work;

T = project delivery schedule; and

$\left(\frac{\Delta r}{\Delta t} + \frac{\Delta q}{\Delta t} + \frac{\Delta rf}{\Delta t} + \left(\frac{\Delta ia}{\Delta t} * t\right)\right) =$ the variables that impact the project scope.

Or this can be described as a change in the overall project scope (number of user stories) due to requirements change, quality factor changes, refactor changes, and information assurance changes for a given time.

$$\left(\frac{\Delta ex}{\Delta t} + \frac{\Delta c}{\Delta t} + \frac{\Delta ep}{\Delta t} + \frac{\Delta p}{\Delta t}\right) = \text{the variables that impact the software development.}$$

Or this can also be described as a change in the software development performance due to team expertise, completeness, experience, and performance.

3.4.4 Baseline results and discussion

After the consolidation of the 46 agile projects that were selected from the 2013–2016 period, the parameters for the projects were collected and analyzed. Using contract types as an initial filter for performance review, it was observed that each contracting approach had varying impact results for the agile projects selected. With a 10% delineation from the baseline schedule and cost parameters as an initial acceptable impact value, the results were tabulated and reviewed. As shown in Table 3-2, firm fixed-price contracts had a more significant impact to the project schedule (>10%) while cost-plus price contracts had a more significant impact on the project cost (>10%). The time-and-material price contract had an effect to both project schedule and cost; however, it was not enough to exceed the 10% delineation significance.

Table 3-2: Agile projects summary per contract type

Type of contract (number of types)	Average project length (days)	Average project cost ($ millions)	Average schedule variance (in days)	Average schedule variance (%)	Average cost variance ($)	Average cost variance (%)
Firm fixed-price contract (20)	501	$6.78M	-57.2	-11.42%	$0.52M	7.71%
Time and material contract (14)	454	$9.30M	-24.5	-5.40%	$0.87M	9.35%
Cost-plus contract (12)	248	$2.03M	-1.75	0.71%	$0.64M	31.43%

Overall, in taking the contract filter off, the 46 selected agile projects also yielded the following average values:

- Average cost ($M): $5.647M
- Average length of project (in days/months): 435 (14.5)

Using assumptions based on real-world experience and agile project management techniques, the following additional values were also utilized for building the model framework:

- The rate of development: 200 user stories per 2-week spin cycle.
- Number of user stories for overall project: 200 x 29 (for 14.5 months) spin cycles = 5,800 user stories
- Average cost per user story: $5.647M / 5,800 user stories = $973.64 per user story

Table 3-3 summarizes the various results using different test cases for the model's behavior. The percentages indicated are based on the original baseline scope of 5,800 user stories; performance capability was set to 100% capability value.

For Test #1, all the parameters for the model were kept at their initial given value (default at 0 or 100%), with only the requirements changing through the development cycle (increasing to 20%). For Test #2, a given requirements change rate (10%) was set, while performance capability was also decreased (to 75%). For Test #3, the requirements change rate (10%) and a decreased performance capability (90%) were applied while quality and refactor change rates were varied (total of 20%). For Test #4, multiple factors with a requirements change rate (10%), decreased performance capability (90%), and quality and refactor impacts were compounded with additional information assurance work (to 5%) and then applied to different timelines during the development cycle (beginning and past the half-way point of the project schedule).

Table 3-3: Baseline model results

Test #	Variable values 1	Variable values 2	Schedule Variance (in days)	Schedule Variance (In %)	Cost Variance (in $)	Cost Variance (In %)
1		Requirements Change – 0%	0		$0.50	
		Requirements Change – 5 %	3	0.67%	-$57,411.00	-1.03%
		Requirements Change – 10%	11	2.40%	-$205,891.50	-3.69%
		Requirements Change –20%	62	14.19%	-$1,215,788.00	-21.76%
2	Requirements @ 10%	Performance @ 90%	19	4.31%	-$369,479.00	-6.61%
		Performance @ 80%	32	7.33%	-$628,291.00	-11.25%
		Performance @ 75%	42	9.44%	-$808,852.00	-14.48%
3	Requirements @ 10% + Performance @ 90%	Quality Deficiency @ 5% Refactor @ 5%	26	5.96%	-$228,301.00	-4.09%
		Quality Deficiency @ 5% Refactor @ 10%	30	6.78%	-$157,712.50	-2.82%
		Quality Deficiency @ 10% Refactor @ 10%	33	7.61%	-$87,123.50	-1.56%
4	Requirements @ 10% + Performance @ 90% + Quality Deficiency @ 5% + Refactor @ 5%	IA work @ 2% (early entry)	26	5.97%	-$223,632.00	-4.00%
		IA work @ 5% (early entry)	27	6.13%	-$201,022.50	-3.60%
		IA work @ 2% (late entry)	26	6.01%	-$227,736.50	-4.08%
		IA work @ 5% (late entry)	32	7.16%	-$224,772.00	-4.02%

The baseline results provided important behavioral patterns that were eventually utilized for analyzing the case studies selected. Figure 3-6 is a graphical representation and comparison of the baseline results for the schedule results. In reviewing the results for the baseline, it was noted that increasing the requirements from the original scope and decreasing the performance capability were the two main contributors for project schedule delays in the software development effort. Although the agile methodology has touted its ability to assimilate new requirements in the development of software for consumer use, for federal managers and software developers using the agile methodology within federal constraints, it is important to understand that contract constraints for schedule or period of performance will play a significant role in the success or failure of delivering software if the requirements are not properly managed. As requirements increase, the agile backlog also increases, and without adding more performance capability to address this change, the probability for delivering software in the prescribed time also diminishes. The results for the baseline model also validated the two projects that were initially discussed at the beginning of the current study. The Healthcare.gov website and the Cover Oregon healthcare initiative website both were reported to have suffered from significant requirements mismanagement, eventually resulting in failure for both efforts despite additional funding and additional resources directed to address the problems

With fewer resources or less capability to perform the necessary work for developing the software, it was also reasonable to assume that the schedule for delivering the software to the customer would also suffer. Generally speaking, this can be applied to all kinds of projects because the capacity to do work is what predicates what kind of

timeline will be accomplished. However, the importance of understanding this causal relationship regarding where the work is performed, in particular within a constrained environment such as the federal software and acquisition construct, cannot be underestimated. For developers and federal managers alike, it is key to understand that a cohesive team with the requisite skill set, knowledge base, experience, and the required amount of resources to handle the effort will play a critical role in the success or failure of delivering software within the appropriate time frame. As the federal government continues to increase its investments in IT, the ability to meet contract requirements within the prescribed time frame can be a critical decision point for federal managers and decision makers in determining the organization that will conduct the development work.

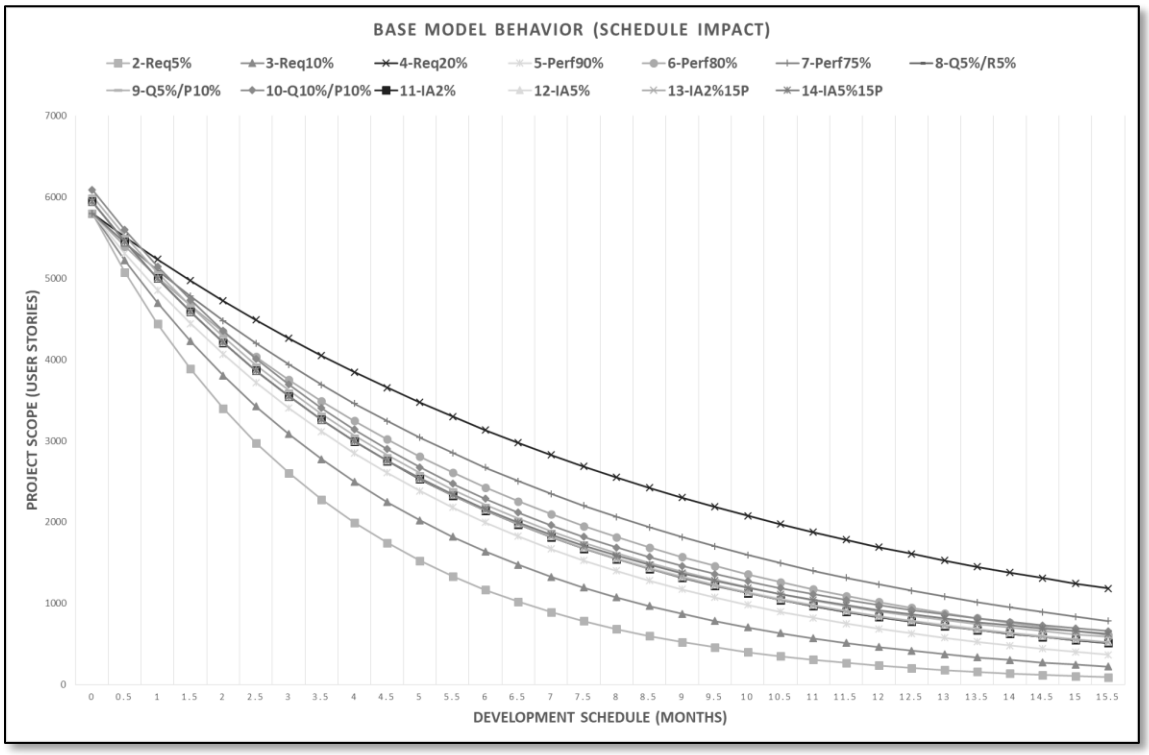

Figure 3-6: Schedule results for base model testing

For the baseline model cost behavior, similar results were obtained regarding the impacts. Changing requirements and decreasing performance resulted in significant

changes in cost calculations and predicted the final project cost. Figure 3-7 is a graphical summary of the cost comparison for the variable changes for the baseline model.

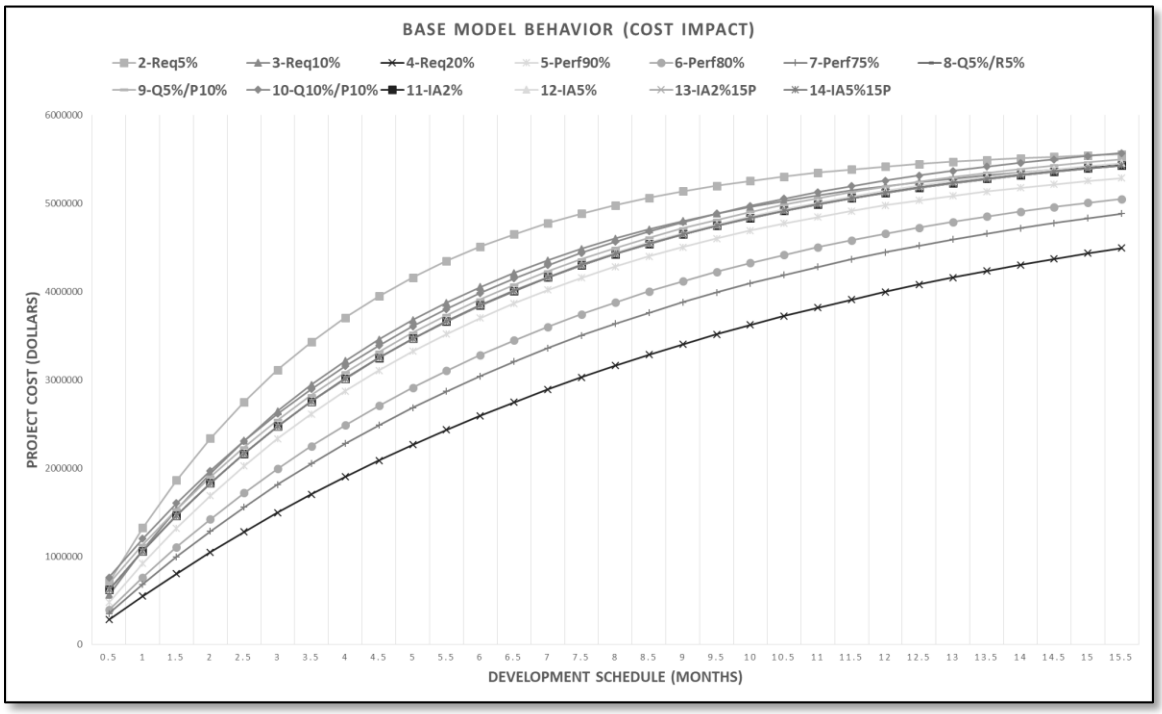

Figure 3-7: Cost results for base model testing

The results for applying the information assurance variables were also noted as a particular point for discussion. The change rate applied (2% and 5% increase in user stories) were evaluated for application at different time points within the development cycle. Although the overall impact was not overly significant in the baseline testing, it was important to note that where in the development cycle the IA impacts were applied had notable differences for additional work required; less in early discovery (before the halfway point of the schedule—5% mark) and more in later discovery (after the halfway point of the schedule—60% mark). This is a key point to bring up for both federal managers and software developers because the results validate that the potential for information assurance rework impacts, such as additional capabilities for control, user access, security integration, and more can play a significant role in the software

development, even when using the agile methodology, and for managers and developers, the need to be cognizant of this effect is paramount. This scenario carries additional significance as information assurance requirements continue to evolve and change due to threats and challenges in the cyber domain.

3.5 Case Study 1 – Purchase Care Program Integrity Enhancement Project

The Purchase Care Program Integrity Enhancement Project is part of the Department of Veterans Affairs' Interagency 21st Century Enrollment Systems Redesign investment; the objective was to provide enhanced program integrity analysis for all Civilian Health and Medical Program of the Department of Veterans Affairs (CHAMPVA) claims. The project utilized the agile methodology for custom software development and has a planned development timeline of 401 days. The project also had an original cost of $975,000 for the total cost. The results of the project are shown in Table 3-4.

Table 3-4: Case Study 1 project results

Original cost ($Millions)	$0.975M	**Final cost ($Millions)**	$1.125M
Original Schedule (Months)	401 days (13.2)	**Final delivery (Months)**	749 days (24.62)

For this particular project, because the contract structure was a firm fixed-price (FFP), the delay in delivery, as well as the increased cost to develop the software, came at the developers' expense. The project had a schedule variance (SV) of -189.13% and a cost variance (CV) of -15%.

Although the agile methodology was utilized for the development, the desired result in both the cost and schedule savings were clearly not achieved for this case study.

The current study's baseline model was utilized to try to understand some of the potential causal factors that created both the schedule and cost variances, which are both critical information for software developers when it comes to firm fixed-price contracts because a delay in schedule and cost overruns are not in the best interest of the developers.

3.6 Case Study 2 – Online Payment Agreement (OPA) Project

The Online Payment Agreement Project is part of the Department of Treasury's Integrated Customer Communication Environment (ICCE) initiative; it had the main objective of allowing online payment for customers in multiple payment formats and agreement systems. The project would allow for secure online transactions for the agency's customers and also provide multilayer authentication for its users to enhance the security for system access. Using agile methodology for developing the software, the project was budgeted at $1.8M with a development schedule of 165 days (5.5 months). The results of this project are shown in Table 3-5.

Table 3-5: Case Study 2 project results

Original cost ($Millions)	$1.8M	**Final cost ($Millions)**	$3.05M
Original Schedule (Months)	165 days (5.5)	**Final delivery (Months)**	165 days (5.5)

This project was awarded a cost-plus incentive fee contract, where the risks for the project was on the government side, so the costs would increase for the effort put forward. For this project, the effort was successful in meeting the schedule requirement at a 0% schedule variance (SV). The cost variance (CV), however, was -69.4%; this was a significant cost increase over the budgeted cost, indicating significant activity or increase in resources involved to develop the product and still achieve the desired development

timeline. Knowing that this effort did not deviate from its schedule delivery while utilizing the agile methodology, the elements of the current study's baseline model were utilized to assess the possible resource factors that could have impacted the increase in cost for the project.

3.7 Case Study 3 – Veterans Affairs (VA) Online Medication Reconciliation Project

The Veterans Affairs Online Medication Reconciliation Report Project was a cross-agency project that was primarily sponsored by the Department of Veterans Affairs (VA), but it had cross-agency requirements with the Department of Defense. As part of an effort to modernize the department's legacy medical systems, the budgeted cost for the project was set at $2.957M with a development timeline of 659 days (21.6 months). The desired end state functionality of the software product was the integration of all medication reports that exist for a particular veteran or patient within the agency's database, thereby providing a comprehensive and continuing patient medical care history for all the VA's patients. The results of the project are shown in Table 3-6.

Table 3-6: Case Study 3 project results

Original cost ($Millions)	$2.957M	Final cost ($Millions)	$4.0M
Original Schedule (Months)	659 days (21.6)	Final delivery (Months)	760 (25)

The contract structure for this project was a time and material contract, with a resulting schedule variance of -5.3% and a cost variance of -35.3%; again, this illustrates the lack of consistency with the expectation that using agile will improve software delivery and cost control for federal agencies. In this example, the implications for

delivering custom software solutions using agile did not demonstrate a significant advantage over previous waterfall software development techniques. Using some assumptions from the current study's baseline model, the possible causes for why this happened were explored as part of the overall study.

3.8 Case Study 4 - Healthcare Associated Infections and Influenza Surveillance System (HAIISS) Phase II Project

The Healthcare Associated Infections and Influenza Surveillance System (HAIISS) Phase II Project was also part of the Department of Veterans Affairs' modernization effort and a subproject within the medical legacy initiative of the agency. The project had a budget of $19.2M and development schedule of 374 days (12.3 months). The objective of this project was to link various medical legacy systems within the agency to one unified database access tool that could help facilitate user information flow and allow a singular point of entry for customer query and database information search. The project results are shown in Table 3-7.

Table 3-7: Case Study 4 project results

Original cost ($Millions)	$19.2M	**Final cost ($Millions)**	$18M
Original Schedule (Months)	374 days (12.3)	**Final delivery (Months)**	368 (12.2)

The project was awarded a firm fixed-price contract structure, and in meeting the delivery schedule and coming in under the budgeted cost, the software developer would receive, in this case, profit for their effort. For this project, using the agile methodology certainly provided the advantages of delivering software on schedule and gave notable cost control results that favored both the government and the software developer.

Utilizing the current study's base model parameters, the intent was to understand possible success factors for the effort and what potential actions can alleviate the impact of constraints to an agile development effort.

4 RESULTS AND ANALYSIS

4.1 Case Study 1 Results – Purchase Care Program Integrity Enhancement Project

For Case Study 1, the narrative for the project was that it suffered from a cost overrun of 15% (+$150k) above its original cost and also had a schedule overrun of almost 90% (+348 days) above its scheduled delivery timeline. With a firm fixed-price contract, these overruns, particularly the cost aspect of the project, are a significant consideration because the software developer would have been liable for the additional cost and failure to deliver on time.

This case study was validated against the baseline model by using the same setup as the baseline model; however, the values for the schedule (13.2 months) and the project cost ($0.975M) were changed to reflect the values for this project. Using the initial values for the rest of the baseline model (default at 0 for additional user stories) for the requirements change, quality change, refactor, and information assurance and using 100% development capability, the Case Study 1 model was established. Initial changes were applied to the requirements change impact, and values were obtained for a variable change from the 0 to 10% addition of user stories because of the requirements change. After obtaining the results for the requirements change, the requirements change value was left at the 10% rate, and the information assurance change rate was adjusted. The information assurance rate change values from 0 to 1.2% were utilized, but only the .5% and the 1.0% values were captured for the discussion to illustrate trending behavior. The information assurance (IA) changes were also factored in at the beginning of the

development cycle (5% mark) for impact behavior and overall development changes. Table 4-1 is a numerical summary of the results of the variable changes.

Table 4-1: Cost and schedule variance for Case Study 1

Changes applied	Cost variance	Schedule variance
+0% Requirement change	0	0
+ 5% increase in user stories due to requirement changes	- 1.28%	4.56%
+ 10% increase in user stories due to requirement changes	- 4.42%	27.6%
+ 10% increase in user stories due to requirement changes + .5% increase in user stories due to IA changes	- 4.5%	58.45%
+ 10% increase in user stories due to requirement changes + 1.0% increase in user stories due to IA changes	- 5.26%	76.36%

Figure 4-1 illustrates the cost variance trending results for the case study as the development timeline extended due to delays. Figure 4-2 illustrates the deviation in the amount of work that still needed to be completed as the timeline extended due to the changing rates. Invariably, the results obtained by changing (increasing) the requirements and the information assurance rate (additional user stories) resulted in both the cost and delivery timeline increasing as well. Although this validation test did not factor in the cost and schedule impacts for both the quality and refactor changes, the indicated values show that for a developer in a firm fixed-price contract, the need to manage deviations or increases in the agile backlog must be carefully managed and addressed at the earliest opportunity because these changes will have a significant impact on the overall end results of the software development, in particular the profit margin for the developer and delivery schedule for the customer.

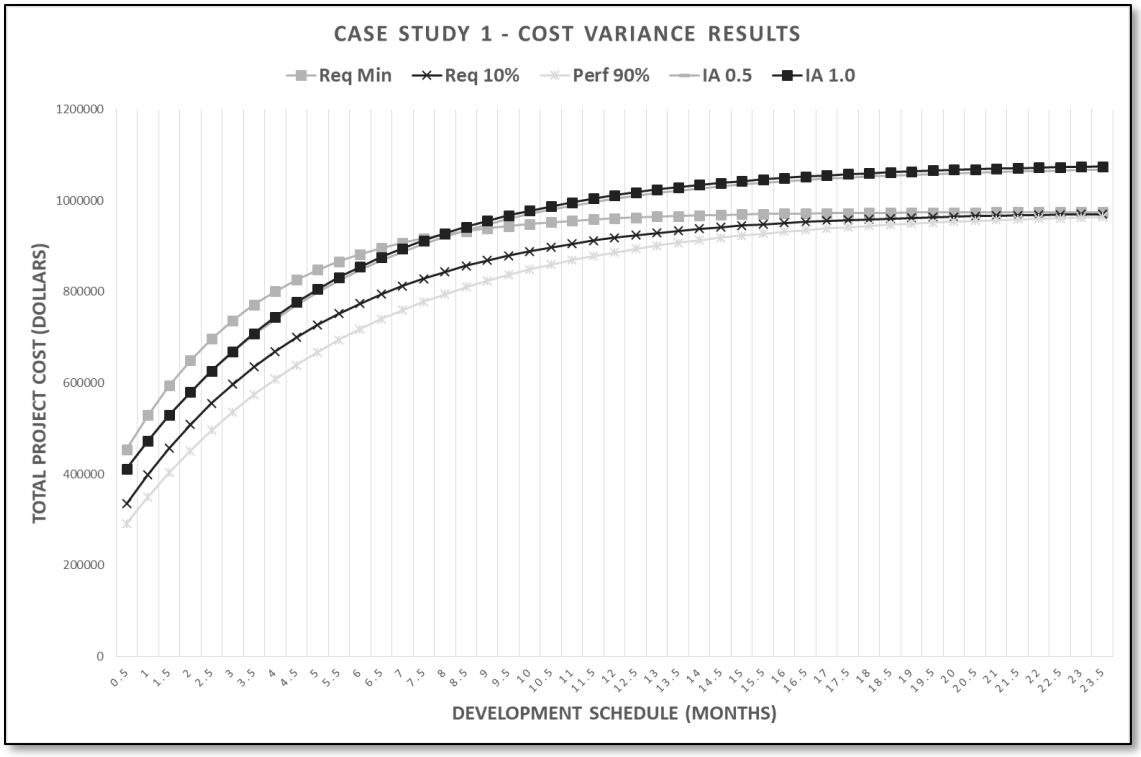

Figure 4-1: Cost results for Case Study 1 testing

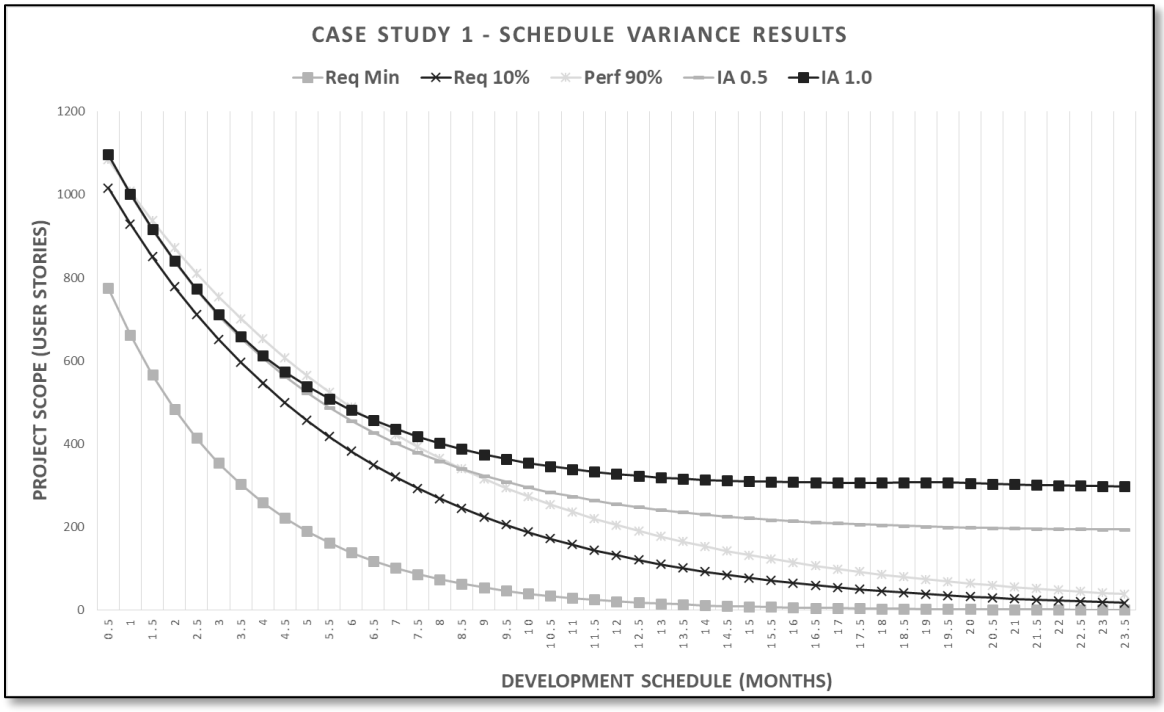

Figure 4-2: Schedule results for Case Study 1 testing

4.2 Case Study 2 Results – Online Payment Agreement (OPA) Project

For Case Study 2, the results of the project indicated that the software was delivered on time (0% schedule variance) but had a significant cost overrun (+70%). The contract was a cost-plus incentive and was designed to support a user interface for online transactions. For this case study, the same set-up was conducted using the baseline model, but adjustments were made to simulate the scheduled delivery timeline (5.5 months) as the case study and cost parameters for the user story ($1,242/user story) increased.

For the case study, the baseline performance capability of developing 200 user stories per sprint cycle of 2 weeks was used. Additional increases to the agile backlog were kept at 0%; however, even at this rate, the results indicated that the developer would not be able to deliver the product on time. The performance was then increased from 200 user stories/spin-cycle to 320 and 440, respectively, with the final value of 650 user stories/spin-cycle as the final performance capability value. Using 200 user stories lefts for the 5.5-month mark to account for quality or refactor rework, the results for the schedule variance were also obtained. Of note, the impact of information assurance rework or user story addition was not taken into consideration for this particular study. Table 4-2 summarizes the results for validating the case study against the baseline parameters, showing the changes applied for the development capability of the agile team.

Table 4-2: Cost and schedule variance for Case Study 2

Changes applied	Cost variance	Schedule variance
Development rate at 200 user stories/spin-cycle	0	-200%
Increase development rate to 320 user stories/spin-cycle	+28.99%	-34.5%
Increase development rate to 440 user stories/spin-cycle	+37.41	+13.5%
Increase development rate to 650 user stories/spin-cycle	+39.74%	+26.5%

Figure 4-3 illustrates the increase in capability to meet the timeline as the performance capabilities increased. Figure 4-4 also shows the increase in cost variance as performance increased to meet the schedule. The results of the study validate that increasing the performance to develop the software will have a corollary increase in cost for the overall effort. For cost-plus contracts, the ability to add resources to the development effort is a significant advantage, especially if requirement changes or other development activities, such as information assurance changes, are added in support of the overall software's capability.

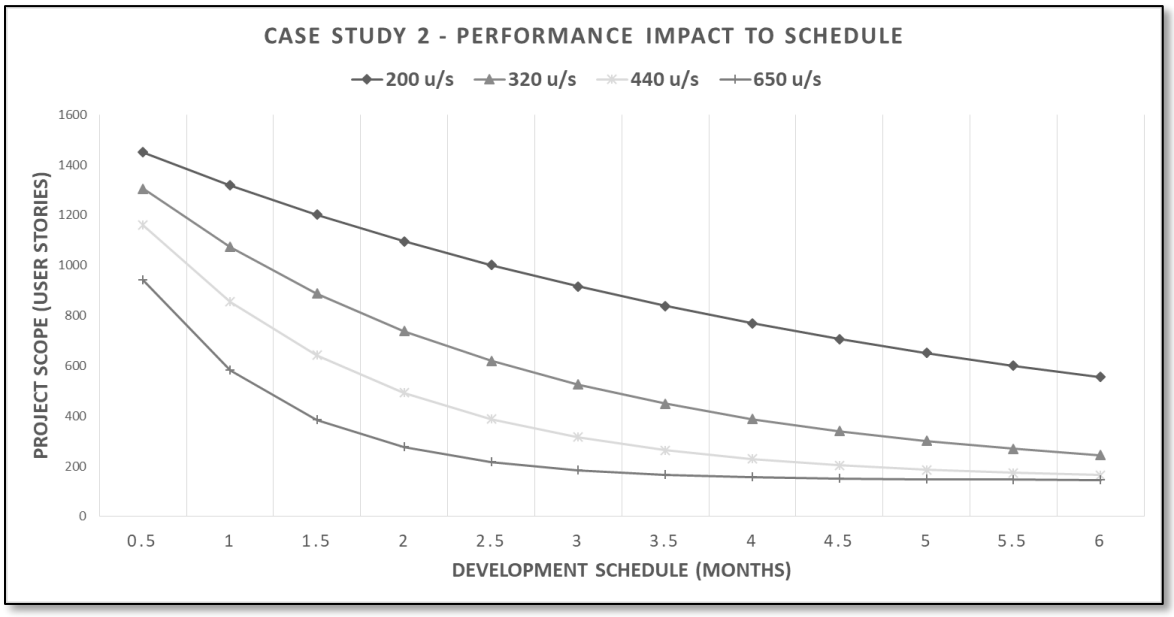

Figure 4-3: Performance impact results for Case Study 2

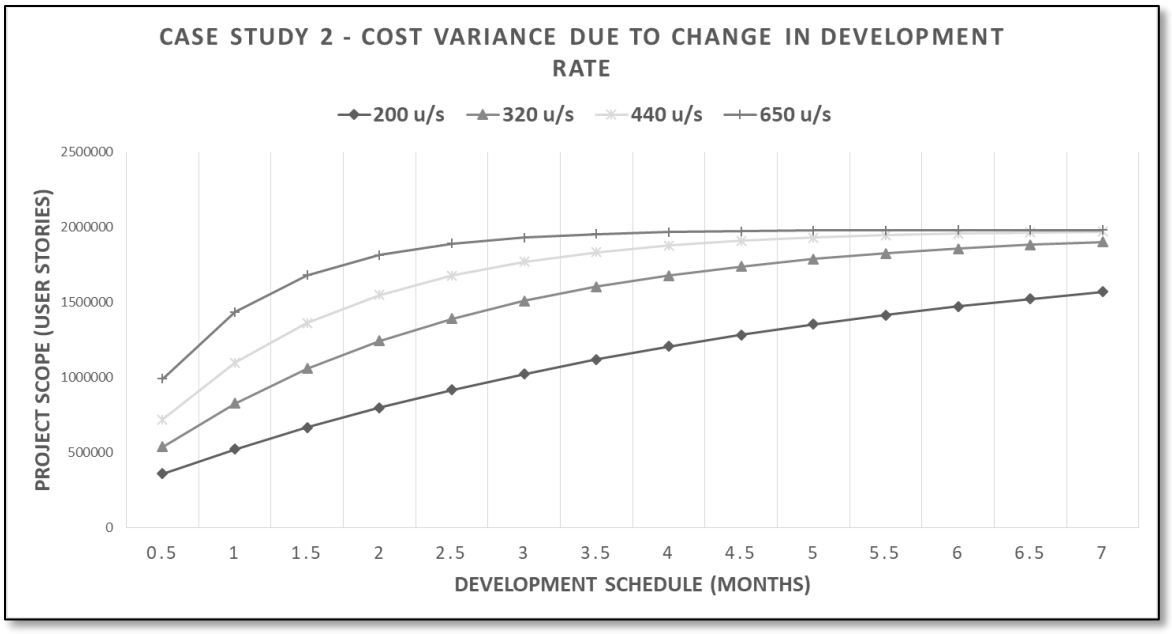

Figure 4-4: Cost behavior results for Case Study 2

4.3 Case Study 3 Results – Veterans Affairs (VA) Online Medication Reconciliation Project (JC3)

For Case Study 3, much like the previous two case studies, the parameters of the baseline model were adjusted to fit the case study's parameters. The original cost ($2.957M) and the original development schedule (659 days) were utilized for the cost and schedule parameters, respectively. Because the contract type awarded to the developer was a time and material contract, the premise for the work conducted is that it will be a "best effort" on the contractor's side to deliver the product, with minimal considerations given for cost and schedule impacts.

For this case study, the capability of the software developer to develop the software was used as the primary variable for consideration. Figure 4-5 illustrates the degradation

of capability by reducing the performance as a percentage taken off of the full development capability.

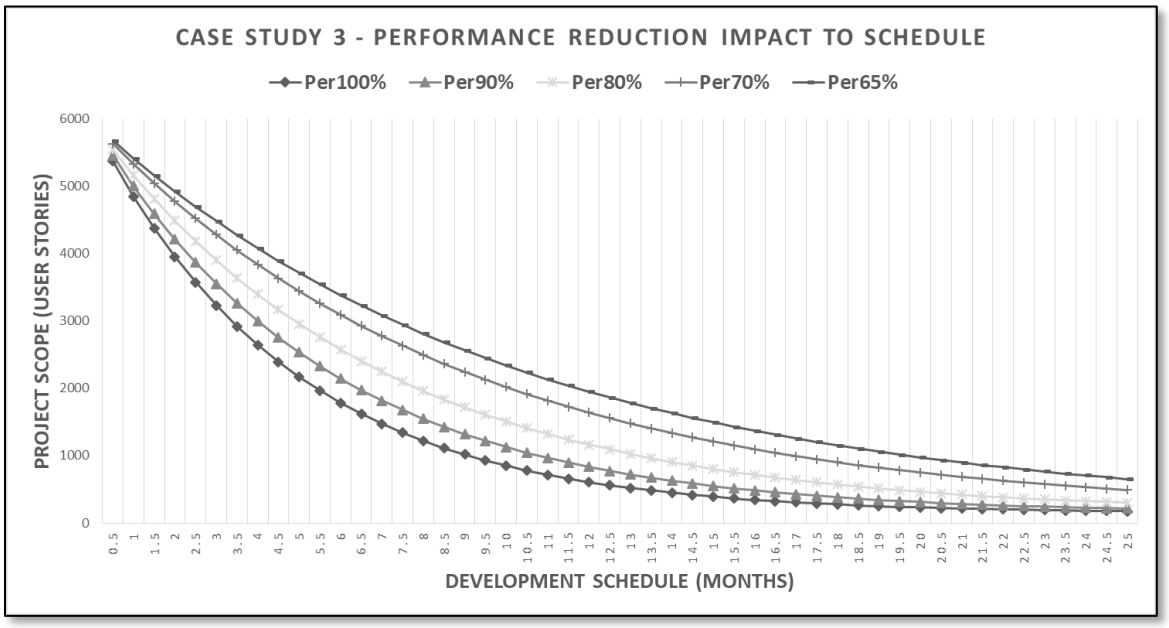

Figure 4-5: Performance reduction results for Case Study 3

For this study, four factors were considered as part of the overall development capability of the software developer. For each factor, a 25% impact value was assigned to assess the overall behavior of the model. The relationship of these factors to the overall agile methodology was based on the findings from the literature review, wherein contributors for the overall software development capability for any agile team were impacted by delays such as schedule slippage due to lack of expertise in using agile methodology, cost slippage due to multiple software re-work due to team experience or overall software delivery delay due to overall poor team coordination. These factors also contribute significantly to the paired-programming techniques that are widely utilized for agile teams and developer constructs. The four factors assessed were as follows:

1. Team expertise: Overall competency of the agile team; is the team familiar with the agile methodology? Does the team adhere to basic agile practices and development cycles? Impact value 25%.
2. Team completeness: The agile team has all the requisite personnel necessary to develop the software. Are there enough developers to conduct quality process and integration rework? Is the team "right-sized" to provide appropriate development tempo? Impact value 25%.
3. Team experience: The overall competency in developing software and requisite skill sets for software delivery. Does the team have enough Java developers? Does the team have the right number of new graduates and experienced developers familiar with C++? Impact value 25%.
4. Team performance: The overall team cohesiveness and ability to communicate and coordinate efforts. Does the team adhere to spin timelines and scrum master assignments? Does the team communicate with other team members on a regular basis? Does the team generally get along and are focused toward the tasks at hand? Impact value 25%.

Because these factors are typically subjective, the impact values utilized were more a generalization and assumption as it pertains to the overall development capability. For the current study, the impact values for the different factors were adjusted to fit the desired overall capability behavior (i.e., team experience and team experience were adjusted to 20% to achieve a total development capability of 90%). This is particularly applicable for a time and material contract because it presumes that the development effort is a "level of effort" capability wherein the software developer will use existing

resources at hand while considering that the full performance capability may not be ready to support the effort. Table 4-3 is a summary of the results obtained from performance adjustments.

Table 4-3: Case Study 3 validation results

Changes applied	Cost variance	Increase in backlog
Development capability at 100%	0	0
Development capability at 90%	-1.04%	31%
Development capability at 80%	-3.14%	91.5%
Development capability at 70%	-7.33%	213%
Development capability at 65%	-10.78%	313.5%

4.4 Case Study 4 Results – Healthcare Associated Infections and Influenza Surveillance System (HAIISS) Phase II Project

The final case study was an application of the baseline model against a successful agile project that satisfied its cost and schedule requirements. The project was completed under the original budget by $1.2M and delivered the final software product 6 days earlier than the original plan. For this analysis, the parameters for the case study were again applied to the baseline model (rate of development and cost per user story), and the development capability impacts were assessed.

The results obtained from the testing showed that at a minimal cost increase, raising the user story development from 200 user stories/spin to 260 user stories/spin resulted in a significant gain in the overall software development timeline. The resulting values indicated a reduction in the agile backlog activities to be completed, allowing for an earlier completion of tasks and early delivery of software. Table 4-4 is a summary of

the adjusted development rate values and resulting variances. As expected, although minimal savings in cost can be achieved by reducing the user story development rate from 200 user stories/spin to 130 and 100 user stories/spin, respectively, this course of action created a significant lag in the ability to meet the scheduled requirements.

Table 4-4: Case Study 4 validation results

Changes applied	Cost variance	Increase in backlog
Development capability at 200 user stories/spin	0	0
Development capability at 260 user stories/spin	1.55%	-77.6%
Development capability at 130 user stories/spin	-6.25%	312%
Development capability at 100 user stories/spin	-14.30%	715%

Figures 4-6 and 4-7 illustrate the behavioral pattern of the tabulated results. Because this project had a firm fixed-price contract, the results provide additional insights for a software developer in understanding what kind of activities can be adjusted to meet the contract's requirements. In this example, actions such as adding additional resources to increase the development rate at a minimal cost can potentially pay significant dividends for the developer, especially in meeting the scheduled requirements. This is especially a key consideration for the developer because this cost can impact the final profit margin for the firm fixed-price contract and must be weighed against the benefit of completing the project on time and satisfying the federal agency's requirements.

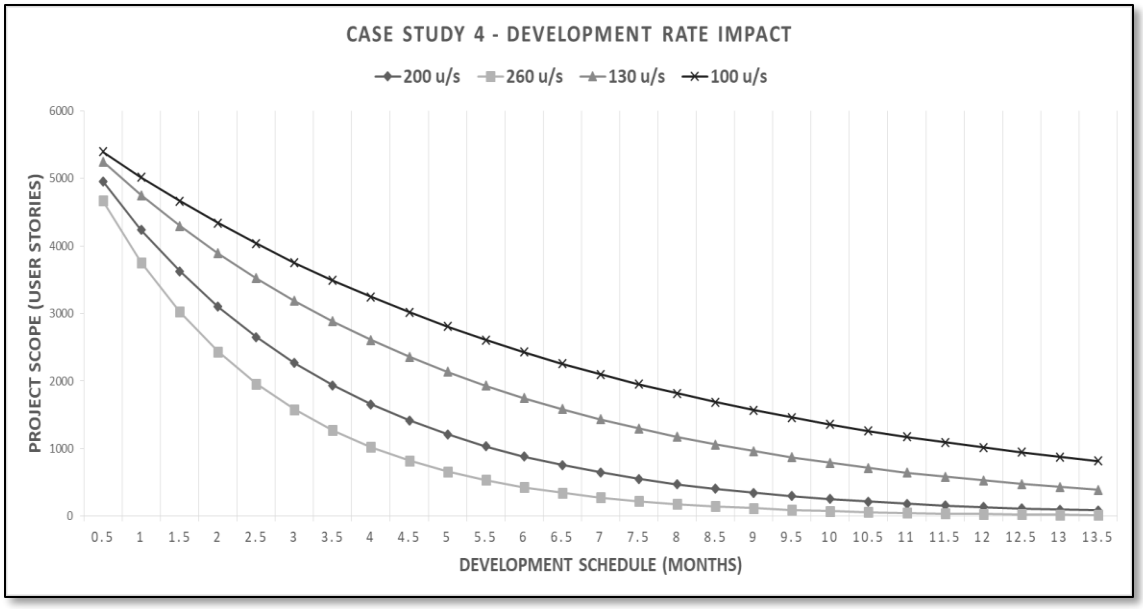

Figure 4-6: Change in development rate results for Case Study 4

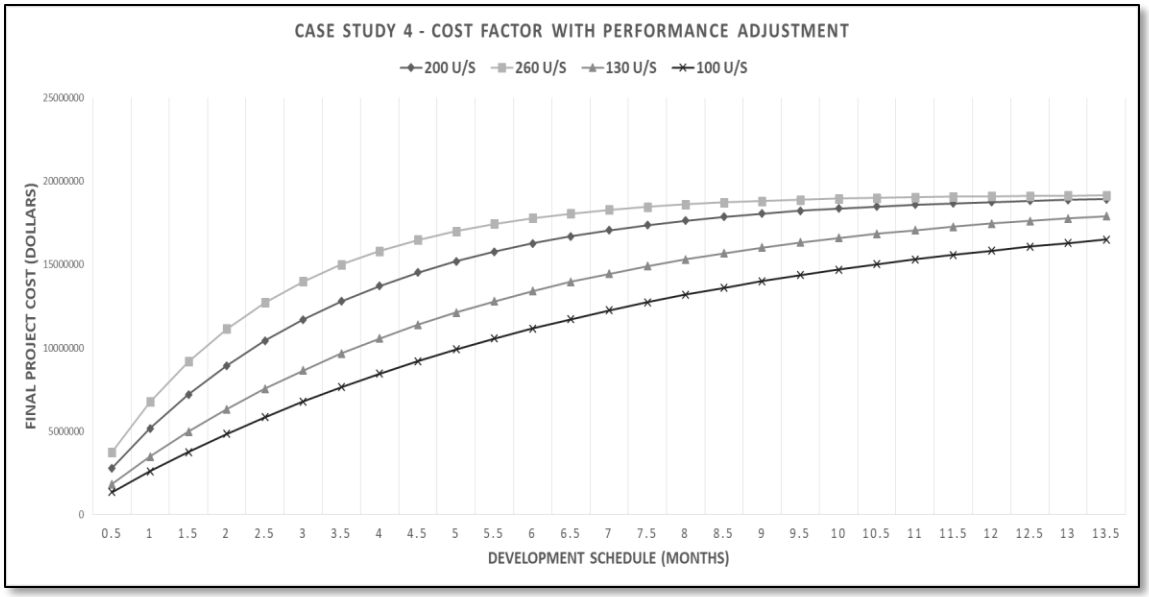

Figure 4-7: Cost behavior results for Case Study 4

4.5 Interpretation of the Results

The overall results of the study validated the two alternate hypothesis, as follows:

1. The agile methodology is impacted by FAR for requirements, contract type, and schedules in its ability to provide software solutions that are responsive to the customer's needs.
2. The use of system dynamics is an effective method for understanding the behavior of software implementation because of its use of metrics and assumptions that can simulate the constraints and challenges for managing the effort needed to successfully work for federal agencies.

For the first hypothesis, the different case studies utilized proved that agile implementation is more than simply a process of developing software: the environment an agile development resides in can influence the success or failure of using this methodology. The results obtained from the case studies provided an extensive understanding of how agile implementation can be challenged if federal constraints are not well-understood and appreciated. Although the results primarily provided percentages for cost and schedule variances, interpolating these results to the various constraints can also provide an understanding and effective implementation of agile strategies for software development and how existing federal acquisition mandates within federal agencies will affect these strategies.

The use of system dynamics to understand the behavior dynamics for software implementation was also determined to be of great utility. Because different values and variables were applied to the model, the results obtained very quickly proved the versatility of system dynamics in assessing and comparing the behavior of different inputs and establishing information that the researcher can utilize. In all, the system dynamics model not only proved that the science behind the methodology is sound, but

that given the opportunity, the model can be used as a framework to define other processes and behavioral studies as variables change and evolve over time.

For Case Study 1, the results provided insights into the potential impacts for a nonstable agile backlog that continues to add user stories because of changing requirements or demands. It is important to understand from a software developers' and federal program manager's perspective that although agile has the ability to adjust work effort because of its iterative capability to deliver in smaller increments, the agile development effort will be impacted in both the cost and schedule parameters if the overall work effort or backlog increases accordingly. Simply put, if additional work effort or user stories are added to the backlog, a commensurate amount of work effort or user stories will also need to be taken out of the backlog to meet the original cost and schedule agreements. This takes on an added layer as it relates to contract types. For firm fixed-price contracts, the increased costs for adjusting or increasing the backlog will impact the software developer's cost parameters, and such, it is paramount for the developer to understand the impact and demands for a nonstable agile backlog.

For Case Study 2, results showed how cost-plus contracts provide a greater amount of flexibility when it comes to increasing resources to address a software need. For an agile effort, this type of contract provides a significant advantage for the developer when addressing schedule and requirement demands because it provides the developer the flexibility to add resources to meet the demands. The results of the study also showed, however, that increasing resources to meet the demands have a corresponding effect on the cost for the federal agency. For federal program managers, this can be a particularly

difficult situation to address, especially if it affects or impacts long-term agency budget plans and if the needed funding is constrained by additional factors outside the agency.

Case Study 3 provided an increased understanding of how agile teams impact software development. Being one of the more debatable topics when it comes to understanding and assessing team effectiveness and productivity, the overall results of this validation study showed the many potential factors that can impact team performance relating to the team's overall capability to address production demands. For agile practitioners and developers, understanding that although no two teams are the same in levels of productivity or skill sets possessed, the requisite impacts of how the team behaves and potential capability the team might possess is valuable in being able to assess how the team will perform against the demand. This is especially useful for software contractors and federal managers because it provides an initial insight or predictable impact that is particularly useful for the selection of contract developers or in developing an agile team.

The last case study illustrated the impact of development rate on cost and schedule constraints. For successful agile projects, "right-sizing" the development team to address the development effort is not only a function of acquiring the right resources to develop the software, but also understanding how the contract's constraints can impact resource availability.

4.6 Utility of the Model

The model developed in this research proved surprisingly versatile and simple in providing an understanding of how processes behave when different constraints are applied. With values obtained from public websites, the model helped create an

understanding of real-world activities and gave results that can be analyzed for use in future efforts, especially in determining the effectiveness of the agile methodology. One surprising aspect of the model was that although the values hold special significance in understanding tangible, real-world impacts and results for future analysis, the model also provided generic behavioral patterns that were not dependent on the researched values; these values that can be adjusted for predicted values to test future prospects. An example for this is in providing user development rates (velocity) or performance factor impacts for future efforts. The model can be utilized for a risk analysis of projected activity success or failure, which can be done by changing the values of the "velocity" or degrading the performance factors to simulate scenarios that will approximate the provided values.

Undoubtedly, the model also has its weak points in that the variables or elements it contains are relatively generic and can have a wide-ranging set of additional variables that can influence or impact the overall results. A good example would be with the information assurance variable; additional factors such as time, interface security, process documentation, and so forth, can add to the overall scope and will require additional work in order to complete the full development effort.

Overall, the model served its primary purpose in providing an overall view of how federal constraints can impact agile implementation in federal agencies. As a means of communicating these interactions, the model can be utilized by both federal managers and software developers in their efforts to manage agile development, hence providing a more effective strategy for managing and navigating work demands against the constrained environment.

5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

Software development using any methodology, whether it is agile, waterfall, a combination of them, or something else is a complicated endeavor. For federal agencies, this becomes even more complex when federal rules and regulations are factored in. The intent of this research was to provide a better understanding of how to manage agile development, not only as a methodology for developing software, but also as a tool to address the existing and wide-ranging needs of the federal government. The agile methodology, as embraced by the commercial world, has proven its worth in delivering software solutions that can rapidly respond to the commercial customers; the federal government, however, remains encumbered by rules and regulations that have not adapted to the rapid and iterative processes that the agile methodology requires.

The research results can provide software developers and federal managers insight as to how federal constraints can impact agile activities and processes. For agile practitioners and developers, this can be a welcome addition to understanding how the federal government conducts business, creating another avenue for discussion and evaluation in seeking business relationships or contract work with federal agencies. For federal managers, the constraints imposed by rules and regulations should not serve as barriers to a successful implementation of agile projects, but rather, they should serve as an understanding of the limitations that exist within the federal environment and provide a more realistic expectation for the results of using the agile process.

5.2 Research Challenges

The largest challenge for the current study came from the limited available data for agile implementation within the federal government. Although the website ITDashboard.gov provided public reports on agile projects and outcomes, the details and determinants for success (or failure) of the projects were limited, sometimes nonexistent. To continue with the present study, various assumptions were taken based on the experience and judgment of the researcher as it pertains to software development using the agile methodology. This included velocity rates, initial requirement numbers, information assurance impacts, and various other factors. Although doing this did not detract from the overall content for understanding the trends and behavioral patterns for impacts of these variables, having real-world, hard data results would have added more realistic scenarios for potential applications to other agile projects that might have similar parameters for constraints.

Due to the time-scope available, the current study was also limited by the lack of a pursuit of deeper causality factors that may have impacted software development. Topics such as performance factors for agile implementation, enterprise software development with integrated waterfall efforts, increased information assurance impacts and reworks, and other notable factors would also have added additional realism to the greater constraints that exist within the federal environment. Again, although this did not prevent achieving the overall end-state desired result, a deeper understanding of these subjects could also have lent to additional information for active, real-world projects that might be facing the same challenges.

5.3 Recommendations for Future Research

The field of software development is rich for future research topics, and there are numerous subjects that can be recommended for future research related to the current study. For the agile methodology itself, additional research in paired programming and the effectiveness of this technique can provide additional insight into developing agile teams and managing outcomes. Future research can also be conducted by initiating sensitivity analysis to the model variables in order to provide additional justification, if not impact validation, for the variables utilized for the model. Data gathering methodologies such as surveys, and historical data reporting, can provide foundational information in order to conduct sensitivity analysis for variable impacts. Additional research can also be conducted for the effectiveness of agile in software integration, particularly in enterprise environments where new and legacy software typically interacts and where understanding the effectiveness of the agile approach in addressing these constraints and challenges could help with the software's outcomes.

With the model's ability to provide process behavior insights, research to the utility of the model as a risk analysis tool for agile software development can also be conducted, in order to provide a more robust correlation of risk factors and identification of event driven consequences for the development effort. With continued refinement of variable data values from reporting agencies, the model's output result will also provide a more defined value, thereby providing robust metrics for the conduct of risk management and validation of activities.

In the federal environment, research on the ongoing impacts of information assurance requirements can pay tremendous dividends in expanding the understanding of

the challenges of using agile in the federal environment. As previously noted under limitations of the model, studies that expand on the addition of time, interface security, process documentation, and other factors will help refine the impact of information assurance constraints to an agile implementation effort. As federal agencies continue to grapple with cybersecurity threats and challenges, new rules and regulations imposed to mitigate these threats also have effects on software development and the delivered solutions. Research into multi-contract impacts on software development is also beneficial as it pertains to agile development.

In all, federal challenges and constraints will continue to be a factor in successfully implementing agile in the federal government; as such, the need for continued studies and engagements with federal managers and software developers will continue to pay dividends when implementing the agile methodology within the federal environment.

References

Abrahamsson, P., Warsta, J., Siponen, M., & Ronkainen, J. (2003). New Directions on Agile Methods: A Comparative Analysis. *Proceedings of the International Conference on Software Engineering.* Portland: IEEE.

Alleman, G., Henderson, M., & Seggelke, R. (2003). Making Agile Work in a Government Contracting Environment. *Agile Development*.

Bird, M. S. (2010, April). Utilizing Agile Software Development as an Effective and Efficient Process to Reduce Development Time and Maintain Quality Software Delivery. Capella University.

Brown, G. A. (2015, March). An Examination of Critical Success Factors of an Agile Project. Capella University.

Cantwell, P. (2013). *The Use of System Dynamics to Understand the Consequences of Project Controls with Dynamic Performance Measures in DoD Project Management.* Washington DC: George Washington University.

Cao, L., & Ramesh, B. (2008). Agile Requirements Engineering Practices: An Empirical Study. *IEEE Software*, 60-67.

Cao, L., Ramesh, B., & Abdel-Hamid, T. (2010, December). Modeling Dynamics in Agile Software development. *ACM Transactions on Management Information Systems*.

Chang , S., & Modigliani, P. (2015). Addressing the Barriers to Agile Development in the Department of Defense: Program Structure, Requirements and Contracting.

Acquisition Research Program Graduate School of Business and Public Policy, Naval Postgraduate School. Monterey CA: Naval Postgraduate School.

Cho, J. (2010). An Exploratory study of Issues and Challenges of Agile Software Development with Scrum. *All Graduate Theses and Dissertation*. Paper 599.

Choopojcharoen, T., & Magzari, A. (2012). *Mathematics Behind System Dynamics.* Worcester: Worcester Polytechnic Institute.

Code of Federal Regulations (CFR). (2017). Retrieved from US Government Publishing Office : https://www.gpo.gov/fdsys/browse/collectionCfr.action?selectedYearFrom=2017&go=Go

Comittee on Oversight and Government Reform. (2016). *Cover Oregon: How Mismanagement and Political Interference Squandered $305 Million Federal Taxpayer Dollars.* Washington DC.

Defense Federal Acquisition Regulation Supplement (DFARS) and Procedures, Guidance, and Information (PGI). (2016, Dec 22). Retrieved from Defense Procurement and Acquisition Policy: http://www.acq.osd.mil/dpap/dars/dfarspgi/current/

Ghadge, A., Dani, S., Chester, M., & Kalawsky, R. (2013). A systems approach for modelling supply chain risks. *Supply Chain Management:An International Journal*, 523-538.

Glaiel, F. (2012). *Agile Project Dynamics:A Strategic Project Management Approach to the Study of Large-Scale Software Development Using System Dynamics.* Massachusets: Massachusets Institute of Technology.

Government Accountability Office. (2016). *Agencies Need to Increase Their Use of Incremental Development Practices.* Washington DC: Government Accountability Office.

Lalsing, V., Kishnah, S., & Pudaruth, S. (2012). People Factors In Agile Software development and Project Management. *International Journal of Software Engineering and Applications*, 117-137.

Martinez, I., & Richardson, G. (2001). Best Practices in System Dynamics Modeling. *The 19th International Conference of the System Dynamics Society.* Atlanta.

Meadows, D. H., Meadows, D. L., Randers, J., & Behren III, W. W. (1972). *The Limits to Growth.* Washington DC: Universe Books, NY.

Miller, S., Hayes, W., & Wrubel, E. (2016). *Agile in Government.* Washington DC: Carnegie Mellon University.

Mueller, T. J. (2014, May). Factors Influencing the Decision to Employ Agile Methods on Federal Information Technology Projects. Robert Morris University.

Office of Management and Budget (OMB). (2017). *ITDashboard*. Retrieved from ITDashboard.gov: https://www.itdashboard.gov/#explore-govwide

Reger, G. (2017, Feb 24). *Agile isn't 100% better for Federal software projects*. Retrieved from www.vergys.com: http://vergys.com/agile-isnt-100-better-for-federal-software-projects/

Rodrigues, A. (1994). The Role of Systems Dynamics in Project Management: A Comparative Analysis with Traditional Models. *International Systems Dynamics Conference,* (pp. 214-225). Stirling UK.

Rodrigues, A., & Williams, T. (1997). System dynamics in software project management: towards the development of a formal integrated framework. *European Journal of Information Systems*, 51-66.

Ronkainen, J., & Abrahamsson, P. (2003). Software Development Under Stringent Hardware Constraints: Do Agile Methods Have a Chance. Genova, IT: LNCS.

Rosenberg, Z., Riasanow, T., & Krcmar, H. (2015). A System Dynamics Model for Business Process Change Projects. *International Conference of the System Dynamics Society*, (pp. 1-27). Cambridge.

Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional. *Informatica Economica*, 64-76.

System Dynamics. (2017, Nov). Retrieved from Business Dictionary: http://www.businessdictionary.com/definition/system-dynamics-SD.html

Tang, V., & Vijay, S. (2001). *System Dynamics: Origin, development, and future prospects of method*. Retrieved from web.MIT.edu: web.mit.edu/esd.83/www/notebook/System%20Dynamics%20final.doc

The Federal Acquisition Requlation. (2017, Jan 01). Retrieved from Acquisition.gov: https://www.acquisition.gov/

The TechFAR Handbook. (n.d.). Retrieved from CIO.gov: https://playbook.cio.gov/techfar/

Timeline: US Healthcare law's technology breakdown. (2013, Oct 30). Retrieved from reuters.com: http://www.reuters.com/article/us-usa-healthcare-delays-idUSBRE99T1C920131030

United States Government Accountability Office (GAO). (2012). *Effective Practices and Federal Challenges in Applying Agile Methods.* Washington DC: United States Government Accountability Office.

United States Government Accountability Office. (2014). *HEALTHCARE.GOV; Ineffective Planning and Oversight Practices Underscore the Needs for Improved Contract Management.* Washington DC: US Government Accountability Office.

Vanroekel, S. (2012, March 6). *Now Live: The Updated IT Dashboard*. Retrieved from The White House: https://obamawhitehouse.archives.gov/blog/2012/03/06/now-live-updated-it-dashboard

Vechnicki, P., & Kelkar, M. (2017). *Agile by the numbers.* Deloitte University Press.

Washington Universtiy in St. Louis. (2017). *What is Systems Dynamics?* Retrieved from Washington University in St. Louis: https://socialsystemdesignlab.wustl.edu/what-is-system-dynamics/

Welby, S. (2013). Thinking About Agile in DoD. *AFEI Agile for Government Summit.*

Yamaguchi, K. (2001). A Step-by-step Systems Dynamics Modeling of Sustainability. *19th International conference of the Systems Dynamics Society.* Atlanta: Osaka Sangyo University, Japan.

Appendix A – Base Model Results

Table A-1: Base Model Schedule Results

Base Model Behavior (Schedule Impact)														
Time (Month)	1-Req0	2-Req5%	3-Req10%	4-Req20%	Req@10%			Req@10%+Perf90%			Req@10%+Perf90%+Q5%/R5%			
					5-Perf90%	6-Perf80%	7-Perf75%	8-Q5%/R5%	9-Q5%/P10%	10-Q10%/P10%	11-IA2%	12-IA5%	13-IA2%15P	14-IA5%15P
0	5799.7	5799.7	5799.7	5799.7	5799.73	5799.76	5799.77	5944.73	6017.23	6089.73	5944.76	5944.88	5944.76	5944.88
0.5	4958.7	5074.7	5219.7	5509.7	5306.73	5393.76	5437.27	5451.73	5524.23	5596.73	5451.77	5451.95	5451.77	5451.95
1	4239.65	4440.33	4697.7	5234.2	4855.63	5016.18	5097.43	5000.63	5073.13	5145.63	5000.7	5000.96	5000.7	5000.96
1.5	3624.85	3885.25	4227.9	4972.48	4442.88	4665.03	4778.83	4587.88	4660.38	4732.88	4587.98	4588.37	4587.98	4588.37
2	3099.21	3399.55	3805.08	4723.84	4065.21	4338.46	4480.14	4210.21	4282.71	4355.21	4210.36	4210.95	4210.36	4210.95
2.5	2649.78	2974.57	3424.54	4487.63	3719.65	4034.75	4200.11	3864.65	3937.15	4009.65	3864.87	3865.75	3864.87	3865.75
3	2265.52	2602.71	3082.06	4263.23	3403.46	3752.3	3937.59	3548.46	3620.96	3693.46	3548.79	3550.11	3548.79	3550.11
3.5	1936.97	2277.34	2773.82	4050.06	3114.14	3489.63	3691.48	3259.14	3331.64	3404.14	3259.59	3261.62	3259.63	3261.62
4	1656.07	1992.63	2496.41	3847.54	2849.41	3245.33	3460.75	2994.41	3066.91	3139.41	2994.93	2998.13	2995.16	2998.13
4.5	1415.89	1743.51	2246.74	3655.15	2607.19	3018.14	3244.44	2752.19	2824.69	2897.19	2752.74	2757.77	2753.2	2757.77
5	1210.55	1525.54	2022.04	3472.37	2385.56	2806.86	3041.65	2530.56	2603.06	2675.56	2531.12	2538.92	2531.82	2538.92
5.5	1034.97	1334.81	1819.8	3298.74	2182.76	2610.36	2851.53	2327.76	2400.26	2472.76	2328.33	2339.19	2329.27	2340.3
6	884.859	1167.92	1637.79	3133.79	1997.2	2427.62	2673.29	2142.2	2214.7	2287.2	2142.78	2155.17	2143.96	2161.02
6.5	756.511	1021.89	1473.98	2977.08	1827.42	2257.67	2506.2	1972.42	2044.92	2117.42	1973	1986.15	1974.4	1997.85
7	646.773	894.118	1326.55	2828.21	1672.06	2099.62	2349.55	1817.06	1889.56	1962.06	1817.64	1831.18	1819.27	1848.9
7.5	552.948	782.316	1193.87	2686.79	1529.92	1952.63	2202.69	1674.92	1747.42	1819.92	1675.5	1689.22	1677.34	1712.94
8	472.727	684.489	1074.45	2552.43	1399.85	1815.92	2065	1544.85	1617.35	1689.85	1545.43	1559.25	1547.48	1588.86
8.5	404.138	598.89	966.977	2424.8	1280.84	1688.79	1935.93	1425.84	1498.34	1570.84	1426.42	1440.29	1428.68	1475.63
9	345.494	523.992	870.249	2303.54	1171.95	1570.56	1814.92	1316.95	1389.45	1461.95	1317.53	1331.42	1319.98	1372.33
9.5	295.354	458.455	783.194	2188.35	1072.31	1460.6	1701.47	1217.31	1289.81	1362.31	1217.89	1231.8	1220.53	1278.09
10	252.484	401.111	704.845	2078.92	981.138	1358.35	1595.12	1126.14	1198.64	1271.14	1126.72	1140.63	1129.54	1192.15
10.5	215.831	350.934	634.33	1974.96	897.719	1263.24	1495.41	1042.72	1115.22	1187.72	1043.3	1057.22	1046.3	1113.78
11	184.492	307.03	570.867	1876.19	821.39	1174.8	1401.93	966.39	1038.89	1111.39	966.97	980.888	970.14	1042.33
11.5	157.697	268.614	513.75	1782.37	751.549	1092.55	1314.3	896.549	969.049	1041.55	897.129	911.048	900.464	977.209
12	134.787	235	462.345	1693.24	687.644	1016.05	1232.14	832.644	905.144	977.644	833.224	847.144	836.719	917.865
12.5	115.2	205.587	416.081	1608.56	629.171	944.912	1155.12	774.171	846.671	919.171	774.751	788.671	778.4	863.802
13	98.4522	179.851	374.443	1528.12	575.669	878.752	1082.91	720.669	793.169	865.669	721.249	735.169	725.047	814.562
13.5	84.1331	157.332	336.969	1451.7	526.714	817.222	1015.21	671.714	744.214	816.714	672.294	686.214	676.236	769.727
14	71.8903	137.628	303.242	1379.1	481.92	760	951.746	626.92	699.42	771.92	627.5	641.42	631.581	728.916
14.5	**61.4227**	**120.387**	**272.888**	**1310.13**	**440.934**	**706.783**	**892.248**	**585.934**	**658.434**	**730.934**	**586.514**	**600.434**	**590.73**	**691.78**
15	52.4729	105.301	245.569	1244.6	403.432	657.292	836.468	548.432	620.932	693.432	549.012	562.932	553.358	658
15.5	44.8209	92.1012	220.982	1182.36	369.117	611.264	784.175	514.117	586.617	659.117	514.697	528.617	519.169	627.283
16	38.2783	80.551	198.854	1123.23	337.719	568.459	735.15	482.719	555.219	627.719	483.299	497.219	487.892	599.363
16.5	32.6845	70.4446	178.938	1067.05	308.99	528.65	689.189	453.99	526.49	598.99	454.57	468.49	459.281	573.996
17	27.9017	61.6016	161.014	1013.68	282.703	491.628	646.101	427.703	500.203	572.703	428.283	442.203	433.107	550.958
Base value	61.4227	61.4227	61.4227	61.4227	61.4227	61.4227	61.4227	61.4227	61.4227	61.4227	61.4227	61.4227	61.4227	61.4227
SV (days)	0	3	11	62	19	32	42	26	30	33	26	27	26	32
SV (%)		0.67%	2.40%	14.19%	4.31%	7.33%	9.44%	5.96%	6.78%	7.61%	5.97%	6.13%	6.01%	7.16%

Table A-2: Base Model Cost Results

Base Model Behavior (Cost Impact)														
Time (Month)	1-Req0	2-Req5%	3-Req10%	4-Req20%	Req@10%			Req@10%+Perf90%			Req@10%+Perf90%+Q5%/R5%			
					5-Perf90%	6-Perf80%	7-Perf75%	8-Q5%/R5%	9-Q5%/P10%	10-Q10%/P10%	11-IA2%	12-IA5%	13-IA2%15P	14-IA5%15P
0	-112	-112	-112	-112	-112	-112	-112	141065.7969	211654.7031	282243.5938	141094.0469	141136.3906	141094.0469	141136.3906
0.5	818719	705777	564599	282243.5	479892.5	395186	352832.5	621070.3125	691659.1875	762248.125	621112.625	621176.1875	621112.625	621176.1875
1	1518820	1323430	1072839	550481.5	919097	762813	683718	1060274.75	1130863.75	1201452.625	1060338.375	1060433.625	1060338.375	1060433.625
1.5	2117406.25	1863876	1530255	805307.5	1320968.5	1104706	993923	1462146.25	1532735.25	1603324.125	1462241.625	1462384.5	1462241.625	1462384.5
2	2629197.25	2336766.5	1941929.75	1047392	1688681.25	1422666.5	1284740.5	1829859	1900448	1971036.875	1830002	1830216.375	1830002	1830216.375
2.5	3066778.75	2750545.75	2312436.75	1277372.5	2025138.25	1718369.75	1557381.5	2166316	2236905	2307493.75	2166530.5	2166852	2166530.5	2166852
3	3440911	3112602.75	2645893	1495854	2332996.5	1993373.75	1812982.75	2474174.25	2544763.25	2615352	2474496	2474978.25	2474496	2474978.25
3.5	3760794	3429402.5	2946003.75	1703411.25	2614686.75	2249127.5	2052608.75	2755864.5	2826453.5	2897042.25	2756347	2757070.5	2756307.75	2757070.5
4	4034293.75	3706602.25	3216103.25	1900590.75	2872433.5	2486978.5	2277258	3013611.25	3084200.25	3154789	3014335	3015420.5	3014115.25	3015420.5
4.5	4268136	3949152	3459192.75	2087911.25	3108271.75	2708180	2487867	3249449.5	3320038.5	3390627.25	3250431.5	3252163.25	3249984	3252118.75
5	4468071.5	4161383	3677973.5	2265865.75	3324063.75	2913897.5	2685312.75	3465241.5	3535830.5	3606419.25	3466473	3469312	3465791	3468341
5.5	4639016	4347085	3874876	2434922.5	3521513.5	3105214.5	2870418.25	3662691.25	3733280.25	3803869	3664164	3668390.75	3663248.5	3666005.75
6	4785174	4509574.5	4052088.5	2595526.5	3702179.75	3283139.5	3043954.75	3843357.5	3913946.5	3984535.25	3845063.5	3850631.75	3843918.5	3846779.5
6.5	4910138.5	4651752.5	4211580	2748100.25	3867489.5	3448609.75	3206645	4008667.25	4079256.25	4149845	4010598.75	4017463.75	4009230	4012143
7	5016983.5	4776158.5	4355122	2893045.25	4018748	3602497	3359167.25	4159925.75	4230514.5	4301103.5	4162075.25	4170193.75	4160489.5	4163428.5
7.5	5108336	4885014	4484309.5	3030743	4157149.25	3745612.5	3502156.75	4298327	4368916	4439505	4300687.5	4310017.5	4298891.5	4301843
8	5186442	4980262	4600578.5	3161556	4283786.5	3878709.5	3636209.5	4424964.5	4495553	4566142	4427528	4438030	4425529	4428487
8.5	5253223	5063604.5	4705221	3285828	4399660	4002489.75	3761884	4540838	4611426.5	4682015.5	4543598.5	4555232.5	4541402.5	4544364
9	5310321	5136529	4799398.5	3403886.75	4505683.5	4117605.5	3879703.5	4646861.5	4717450	4788039	4649812.5	4662541	4647426	4650389
9.5	5359139	5200337.5	4884159	3516042.5	4602695.5	4224663	3990159.5	4743873.5	4814462	4885051	4747008.5	4760795	4744438	4747402
10	5400879	5256170.5	4960443	3622590.5	4691461.5	4324227	4093712.25	4832639.5	4903228	4973817	4835952.5	4850761.5	4833204	4836168.5
10.5	5436566.5	5305024	5029098.5	3723811	4772682	4416821	4190792.75	4913860	4984448.5	5055037.5	4917344.5	4933143	4914424.5	4917389
11	5467079.5	5347771	5090889	3819970.5	4846999	4502933.5	4281806	4988177	5058765.5	5129354.5	4991828	5008582	4988741.5	4991706
11.5	5493168	5385175	5146500	3911322	4914999	4583018	4367130.5	5056177	5126765.5	5197354.5	5059988.5	5077666.5	5056741.5	5059706
12	5515473.5	5417903	5196550	3998105.75	4977219.5	4657497	4447122	5118397.5	5188986	5259575	5122364.5	5140935	5118962	5121926.5
12.5	5534545	5446540	5241595	4080550.5	5034151	4726762	4522114.5	5175329	5245917.5	5316506.5	5179446	5198880	5175893.5	5178858
13	5550851	5471597.5	5282135.5	4158873	5086243	4791178.5	4592420	5227421	5298009.5	5368598.5	5231683	5251952	5227985.5	5230950
13.5	5564792.5	5493523	5318622	4233279	5133907.5	4851086	4658331	5275085.5	5345674	5416263	5279488	5300563.5	5275650	5278615
14	5576712.5	5512707.5	5351459.5	4303965.5	5177520	4906800	4720123	5318698	5389286.5	5459875.5	5323236	5345091.5	5319262.5	5322227
14.5	**5586904.5**	**5529494**	**5381013.5**	**4371117**	**5217426**	**4958614**	**4778053**	**5358604**	**5429192.5**	**5499781.5**	**5363273**	**5385882.5**	**5359168.5**	**5362133**
15	5595618	5544182.5	5407612.5	4434911	5253939.5	5006801	4832362	5395117.5	5465706	5536295	5399913	5423251	5395682	5398647
15.5	5603068.5	5557034.5	5431551	4495515.5	5287350	5051615	4883277	5428528	5499116.5	5569705.5	5433446	5457488.5	5429092.5	5432057
16	5609438.5	5568280	5453096	4553090	5317920	5093292	4931009.5	5459098	5529686.5	5600275.5	5464134.5	5488858	5459662.5	5462627
16.5	5614885	5578120	5472486.5	4607785	5345892	5132051.5	4975759	5487070	5557658.5	5628247.5	5492221	5517602.5	5487634.5	5490599
17	5619541.5	5586730	5489938	4659746	5371486	5168098	5017711.5	5512664	5583252.5	5653841.5	5517926	5543943.5	5513228.5	5516193
Base value	$5,586,905.00	$5,586,905.00	$5,586,905.00	$5,586,905.00	$5,586,905.00	$5,586,905.00	$5,586,905.00	$5,586,905.00	$5,586,905.00	$5,586,905.00	$5,586,905.00	$5,586,905.00	$5,586,905.00	$5,586,905.00
CV ($)	$0.50	-$57,411.00	-$205,891.50	-$1,215,788.00	-$369,479.00	-$628,291.00	-$808,852.00	-$228,301.00	-$157,712.50	-$87,123.50	-$223,632.00	-$201,022.50	-$227,736.50	-$224,772.00
CV (%)		-1.03%	-3.69%	-21.76%	-6.61%	-11.25%	-14.48%	-4.09%	-2.82%	-1.56%	-4.00%	-3.60%	-4.08%	-4.02%

Appendix B – Case Study 1 Results

Table B-1: Schedule Results

Case Study 1 - Schedule Results											
Time (Month)	**Req Min**	**Req 5%**	**Req 10%**	**Perf 90%**	**Perf 80%**	**IA 0**	**IA 0.2**	**IA 0.5**	**IA 0.7**	**IA 1.0**	**IA 1.2**
0	1449.69995	1449.69995	1449.72998	1449.76001	1449.77502	1594.7	1594.729	1594.7725	1594.8015	1594.845	1594.8739
0.5	1239.44995	1268.44995	1326.47998	1348.26001	1359.15002	1449.7	1449.7434	1449.8087	1449.8522	1449.9175	1449.9609
1	1059.68616	1109.8562	1213.70618	1253.86499	1274.18909	1319.2	1319.2651	1319.363	1319.4283	1319.5263	1319.5914
1.5	905.98822	971.08673	1110.51819	1166.07764	1194.53821	1201.75	1201.8479	1201.9946	1202.0925	1202.2394	1202.3373
2	774.57642	849.66339	1016.1012	1084.43542	1119.86548	1096.0449	1096.1918	1096.412	1096.5588	1096.7791	1096.9259
2.5	662.2193	743.41797	929.70966	1008.50818	1049.85986	1000.9105	1001.1307	1001.4611	1001.6813	1002.0116	1002.2319
3	566.15399	650.45319	850.66138	937.89581	984.22955	915.28949	915.61981	916.1153	916.44562	916.9411	917.27142
3.5	484.01816	569.10901	778.33221	872.22632	922.70117	838.23053	838.72601	839.46924	839.96478	840.70801	841.20349
4	413.79202	497.93286	712.151	811.15369	865.01831	768.8775	769.62073	770.7356	771.47882	772.59369	773.33691
4.5	353.74869	435.65375	651.59521	754.35614	810.94061	706.45978	707.46826	709.24689	710.36176	712.03406	713.14893
5	302.41162	381.15952	596.18665	701.53442	760.24274	650.28381	651.54865	654.46454	656.13678	658.64526	660.31757
5.5	258.51843	333.47708	545.48779	652.41022	712.7135	599.7254	601.2381	605.57928	608.50494	612.26758	614.776
6	220.98975	291.75494	499.09842	606.72473	668.15485	554.2229	555.97516	561.69415	566.26227	573.03613	576.79877
6.5	188.90273	255.24806	456.6521	564.23718	626.3811	513.27057	515.25446	522.3053	528.46112	538.70673	545.53394
7	161.46832	223.30455	417.81372	524.72375	587.2182	476.41354	478.62128	486.95959	494.65024	508.25177	518.04144
7.5	138.01192	195.35399	382.27661	487.97629	550.50299	443.24219	445.66632	455.24921	464.42349	481.26917	493.92249
8	117.95669	170.89723	349.76016	453.80115	516.08252	413.38797	416.0213	426.80725	437.41571	457.39737	472.81891
8.5	100.80946	149.49757	320.0076	422.01828	483.81326	386.51917	389.35474	401.30362	413.29849	436.31158	454.40909
9	86.14859	130.77287	292.784	392.46021	453.56088	362.33725	365.36829	378.44141	391.77643	417.71991	438.40417
9.5	73.61354	114.38876	267.87442	364.97119	425.19925	340.57352	343.79352	357.95337	372.58389	401.36011	424.54486
10	62.89608	100.05267	245.08212	339.4064	398.61023	320.98615	324.38882	339.59918	355.48203	386.99652	412.59848
10.5	53.73265	87.50858	224.22719	315.63116	373.68301	303.35754	306.93677	323.1626	340.25604	374.41757	402.35614
11	45.89791	76.53251	205.14493	293.52017	350.31375	287.49179	291.24173	308.44919	326.71283	363.43314	393.63043
11.5	39.19921	66.92844	187.68466	272.95697	328.40509	273.21262	277.12756	295.28394	314.6788	353.87259	386.2533
12	33.47183	58.52489	171.70851	253.83318	307.86572	260.36136	264.43579	283.50946	303.99783	345.58267	380.07401
12.5	28.57491	51.17178	157.09033	236.04805	288.61005	248.79521	253.02383	272.98425	294.52966	338.42584	374.95746
13	24.38805	44.73781	143.71471	219.50789	270.55786	238.3857	242.76335	263.5809	286.14816	332.27863	370.78253
13.5	**20.80828**	**39.10808**	**131.476**	**204.12553**	**253.63393**	**229.01712**	**233.53886**	**255.18498**	**278.73999**	**327.03027**	**367.44073**
14	17.74758	34.18207	120.2776	189.81995	237.76775	220.5854	225.24644	247.69353	272.20337	322.58148	364.83487
14.5	15.13068	29.87181	110.03105	176.51575	222.8932	212.99687	217.79253	241.01387	266.44672	318.84308	357.99689
15	12.89323	26.10034	100.65546	164.14284	208.94832	206.16719	211.09297	235.06279	261.38788	315.73517	351.16721
15.5	10.98021	22.80029	92.0768	152.63603	195.87498	200.02046	205.07207	229.76538	256.95297	313.18619	345.02048
16	9.34458	19.91276	84.22732	141.93471	183.61873	194.48842	199.66164	225.05434	253.07565	311.13193	339.48843
16.5	7.94612	17.38616	77.04504	131.98248	172.12849	189.50957	194.80035	220.86913	249.6964	309.51498	334.50958
17	6.75043	15.17539	70.47327	122.72691	161.3564	185.02861	190.43303	217.15535	246.7617	308.28384	330.02863
17.5	5.72812	13.24097	64.46009	114.11923	151.25757	180.99574	186.51003	213.86409	244.22359	307.39249	325.99576
18	4.85404	11.54835	58.95803	106.11408	141.7899	177.36618	182.98665	210.95142	242.03893	306.79965	322.36618
18.5	4.10671	10.0673	53.92365	98.6693	132.91397	174.09956	179.82268	208.37779	240.16905	306.46863	319.09958
19	3.46773	8.77139	49.31718	91.74564	124.5928	171.15961	176.98195	206.10773	238.57928	306.36633	316.15961
19.5	2.92141	7.63747	45.10228	85.30665	116.79169	168.51364	174.43192	204.10931	237.23849	306.4635	313.51367
20	2.45431	6.64528	41.24563	79.31838	109.47814	166.13228	172.14326	202.35394	236.11879	306.7338	311.13229
20.5	2.05493	5.77712	37.7168	73.7493	102.6217	163.98904	170.08968	200.81581	235.19519	307.15387	308.98907
21	1.71347	5.01748	34.48793	68.57005	96.19378	162.06013	168.24742	199.47185	234.44524	307.06015	307.06015
21.5	1.42151	4.3528	31.5335	63.75334	90.1676	160.32413	166.59515	198.30127	233.84888	305.32416	305.32416
22	1.1719	3.7712	28.8302	59.2738	84.51807	158.76172	165.11371	197.28546	233.38815	303.76172	303.76172
22.5	0.95847	3.2623	26.35669	55.10784	79.22163	157.35555	163.78581	196.40767	233.04692	302.35556	302.35556
23	0.77599	2.81701	24.09342	51.23349	74.25621	156.08998	162.5959	195.65288	232.81084	301.09	301.09
23.5	0.61997	2.42738	22.02253	47.63034	69.60114	154.95099	161.53004	195.00761	232.66696	299.95102	299.95102
24	0.48658	2.08646	20.12766	44.27942	65.237	153.92589	160.57564	194.45979	232.60384	298.9259	298.9259
24.5	0.37252	1.78815	18.39386	41.16306	61.14562	153.0033	159.72139	193.99857	232.61115	298.00333	298.00333
25	0.27501	1.52713	16.80743	38.26485	57.30996	152.17297	158.95714	193.61423	232.67972	297.17297	297.17297
Base value	20.80828	20.80828	20.80828	20.80828	20.80828	20.80828	20.80828	20.80828	20.80828	20.80828	20.80828
SV	0	18.2998	110.66772	183.31725	232.82565	208.20884	212.73058	234.3767	257.93171	306.22199	346.63245
SV (%)	0.00%	4.56%	27.60%	45.72%	58.06%	51.92%	53.05%	58.45%	64.32%	76.36%	86.44%

Table **B**-2: Cost Results

Case Study 1 - Cost Results											
Time (Month)	**Req Min**	**Req 5%**	**Req 10%**	**Perf 90%**	**Perf 80%**	**IA 0**	**IA 0.2**	**IA 0.5**	**IA 0.7**	**IA 1.0**	**IA 1.2**
0	-9	-9	-9	-9	-9	97491.898	97511.406	97540.641	97560.148	97589.398	97608.898
0.5	141367.3125	121867.125	97491.9375	82866.8125	68241.625	187680.28	187709.53	187753.41	187782.66	187826.53	187855.78
1	262244.0625	228508.75	185242.75	158698.125	131714.75	269526.22	269570.09	269635.91	269679.78	269745.59	269789.47
1.5	365593.6875	321820.188	264218.438	228083.813	190744.75	343801.34	343867.16	343965.88	344031.69	344130.41	344196.22
2	453957.625	403467.625	335296.563	291571.688	245642.5625	411206.03	411304.75	411452.81	411551.53	411699.63	411798.34
2.5	529508.75	474909.188	399266.938	349663.063	296697.5625	472375.84	472523.91	472746.03	472894.13	473116.22	473264.31
3	594105	537420.563	456840.25	402816.75	344178.75	527887.38	528109.5	528442.69	528664.81	528998	529220.13
3.5	649334.8125	592118	508656.219	451452.313	388336.25	578264.19	578570.25	579097.13	579430.31	579930.06	580263.25
4	696556.25	639978.25	555290.625	495953.875	429402.6875	623981	624329.06	625230.44	625730.19	626479.88	626979.63
4.5	736930.625	681856	597261.5	536672.75	467594.4688	665469.13	665838.19	667312.63	668092.94	669217.38	669967.06
5	771450.6875	718499	635035.375	573930.625	503112.875	703119.5	703499	705260	706820.19	708741.94	709866.38
5.5	800965.3125	750561.625	669031.813	608021.5	536145	737287.31	737672.13	739576.38	741526.44	744973.56	747407.63
6	826200.375	778616.438	699628.625	639214.75	566864.75	768294.56	768681.94	770657.88	772802.88	777012.75	780374.81
6.5	847776.3125	803164.375	727165.75	667756.5	595434.25	796433.63	796822.31	798834.06	801076.56	805667.75	809493.81
7	866223.75	824643.813	751949.188	693872.125	622003.875	821969.88	822359.25	824388.88	826680.13	831462	835520.06
7.5	881996.3125	843438.375	774254.25	717768	646713.625	845143.94	845533.63	847572.19	849887.88	854765.06	858939.13
8	895481.8125	859883.563	794328.875	739632.75	669693.625	866174.5	866564.38	868607.38	870935.25	875860.13	880092.13
8.5	907012	874273.125	812396	759638.938	691065.125	885259.63	885649.56	887694.88	890028.75	894977.5	899238.5
9	916870.25	886864	828656.375	777944.625	710940.5	902579.5	902969.5	905015.88	907352.88	912313.5	916589
9.5	925299.0625	897881	843290.75	794694.375	729424.6875	918297.19	918687.19	920734.13	923072.63	928039.25	932322
10	932505.6875	907520.875	856461.688	810020.313	746615	932561	932951	934998.25	937337.5	942307.06	946593.5
10.5	938667.375	915955.75	868315.5	824043.625	762601.9375	945505.44	945895.44	947942.81	950282.5	955253.56	959541.75
11	943935.625	923336.313	878983.938	836874.875	777469.8125	957252.56	957642.56	959690.06	962029.88	967001.69	971290.81
11.5	948439.9375	929794.25	888585.563	848615.5	791296.875	967913	968303	970350.5	972690.44	977662.63	981952.19
12	952291.125	935445	897227	859358.188	804156.125	977587.38	977977.38	980024.88	982364.88	987337.25	991627.06
12.5	955583.9375	940389.375	905004.313	869187.75	816115.1875	986366.88	986756.88	988804.38	991144.38	996116.88	1000406.8
13	958399.25	944715.688	912003.875	878181.813	827237.125	994334.25	994724.25	996771.81	999111.81	1004084.3	1008374.3
13.5	**960806.375**	**948501.25**	**918303.5**	**886411.375**	**837580.5**	**1001564.7**	**1001954.7**	**1004002.3**	**1006342.3**	**1011314.8**	**1015604.8**
14	962864.4375	951813.563	923973.125	893941.375	847199.875	1008126.3	1008516.3	1010563.8	1012903.8	1017876.4	1022166.4
14.5	964624.125	954711.875	929075.813	900831.375	856145.875	1014080.9	1014470.9	1016518.5	1018858.5	1023831.1	1028121.1
15	966128.625	957247.875	933668.25	907135.688	864465.6875	1019484.8	1019874.8	1021922.3	1024262.3	1029234.9	1033524.9
15.5	967414.9375	959466.875	937801.438	912904.188	872203.125	1024388.8	1024778.8	1026826.3	1029166.3	1034138.9	1038428.9
16	968514.8125	961408.563	941521.25	918182.313	879398.875	1028839.1	1029229.1	1031276.7	1033616.7	1038589.3	1042879.3
16.5	969455.125	963107.5	944869.125	923011.813	886090.9375	1032877.9	1033267.9	1035315.4	1037655.4	1042628	1046918
17	970259.125	964594.063	947882.25	927430.813	892314.625	1036543	1036933	1038980.6	1041320.6	1046293.1	1050583.1
17.5	970946.5625	965894.813	950594	931474.188	898102.5625	1039869.1	1040259.1	1042306.6	1044646.6	1049619.1	1053909.3
18	971534.3125	967032.938	953034.625	935173.875	903485.375	1042887.5	1043277.5	1045325.1	1047665.1	1052637.6	1056927.6
18.5	972036.8125	968028.813	955231.125	938559.125	908491.4375	1045626.8	1046016.8	1048064.3	1050404.3	1055376.9	1059666.9
19	972466.5	968900.188	957208.063	941656.563	913147	1048112.6	1048502.6	1050550.1	1052890.3	1057862.8	1062152.8
19.5	972833.875	969662.688	958987.25	944490.75	917476.75	1050368.5	1050758.5	1052806	1055146	1060118.6	1064408.6
20	973147.9375	970329.875	960588.5	947084.063	921503.375	1052415.8	1052805.8	1054853.3	1057193.3	1062165.9	1066455.9
20.5	973416.5	970913.625	962029.625	949456.938	925248.125	1054273.6	1054663.6	1056711.1	1059051.3	1064023.8	1068313.8
21	973646.125	971424.438	963326.688	951628.063	928730.75	1055959.8	1056349.8	1058397.3	1060737.3	1065709.8	1069999.9
21.5	973842.4375	971871.375	964494	953614.688	931969.625	1057489.8	1057879.8	1059927.3	1062267.3	1067239.9	1071529.9
22	974010.25	972262.438	965544.625	955432.438	934981.75	1058878.3	1059268.3	1061315.8	1063655.8	1068628.4	1072918.4
22.5	974153.75	972604.625	966490.188	957095.688	937783	1060138.4	1060528.4	1062575.9	1064916	1069888.5	1074178.5
23	974276.5	972904.063	967341.125	958617.563	940388.1875	1061281.9	1061671.9	1063719.4	1066059.5	1071032	1075322
23.5	974381.375	973166.063	968107	960010.063	942811	1062319.6	1062709.6	1064757.1	1067097.3	1072069.8	1076359.8
24	974471.0625	973395.313	968796.313	961284.188	945064.25	1063261.4	1063651.4	1065698.9	1068039	1073011.5	1077301.5
24.5	974547.8125	973595.875	969416.688	962450.063	947159.75	1064116.1	1064506.1	1066553.6	1068893.6	1073866.1	1078156.3
25	974613.375	973771.375	969975	963516.813	949108.5625	1064891.6	1065281.6	1067329.1	1069669.3	1074641.8	1078931.8
Base value	960806.4	960806.4	960806.4	960806.4	960806.4	960806.4	960806.4	960806.4	960806.4	960806.4	960806.4
CV ($)	-$0.03	-$12,305.15	-$42,502.90	-$74,395.03	-$123,225.90	$40,758.29	$41,148.29	$43,195.85	$45,535.85	$50,508.35	$54,798.35
CV(%)	0.00%	-1.28%	-4.42%	-7.74%	-12.83%	4.24%	4.28%	4.50%	4.74%	5.26%	5.70%

Appendix C – Case Study 2 Results

Table C-1: Schedule Impact Results

Case Study 2 - Performance Impact to Schedule								
Time (Month)	**200 u/s**	**260 u/s**	**320 u/s**	**380 u/s**	**440 u/s**	**540 u/s**	**650 u/s**	**850 u/s**
0	1594.729	1594.62903	1594.52905	1594.42908	1594.3291	1594.229	1594.02905	1593.8291
0.5	1449.74341	1377.14343	1304.54346	1231.94348	1159.34351	1086.74341	941.54352	796.34344
1	1319.26514	1192.29016	1072.56519	960.09027	854.86523	756.89026	582.69025	437.4902
1.5	1201.8479	1035.17908	886.99792	756.21667	641.74786	542.50409	385.34161	276.02908
2	1096.19177	901.65588	738.56689	603.33594	492.59183	403.18088	276.83087	203.40587
2.5	1001.13074	788.19287	619.85626	488.71201	388.22171	312.66235	217.19644	170.77678
3	915.61981	691.79706	524.93915	402.79916	315.22137	253.8877	184.46725	156.17078
3.5	838.68573	609.89185	449.04224	338.4068	264.16888	215.73746	166.53049	149.67337
4	769.39508	540.22876	388.28723	290.0813	228.40721	190.92107	156.659	146.75577
4.5	707.00861	480.99335	339.66452	253.82159	203.36154	174.78107	151.22658	145.44598
5	650.84821	430.63226	300.75696	226.61899	185.82332	164.28535	148.23717	144.85812
5.5	**600.29767**	**387.81989**	**269.62628**	**206.21313**	**173.54344**	**157.46083**	**146.59224**	**144.59439**
6	554.79901	351.42661	244.71935	190.9068	164.94597	153.0237	145.68712	144.47607
6.5	513.84863	320.491	224.79265	179.42607	158.92696	150.13899	145.1891	144.42305
7	476.99255	294.19504	208.85071	170.81503	154.71327	148.26364	144.9151	144.39928
7.5	443.82172	271.84311	196.09686	164.35652	151.76349	147.04451	144.76436	144.38863
8	413.96771	252.84383	185.89363	159.51251	149.69853	146.252	144.68143	144.38387
8.5	387.09906	236.69434	177.73099	155.87944	148.25302	145.73685	144.63579	144.38173
9	362.91718	222.96722	171.20082	153.15462	147.24115	145.40196	144.61069	144.38077
9.5	341.1535	211.29916	165.97667	151.11098	146.53282	145.18428	144.59689	144.38034
10	321.56616	201.38132	161.79735	149.57823	146.03697	145.04279	144.58929	144.38016
10.5	303.93753	192.95111	158.45387	148.42868	145.68988	144.95081	144.5851	144.38007
11	288.07178	185.78546	155.77911	147.56651	145.44693	144.89104	144.58281	144.38004
11.5	273.7926	179.69464	153.6393	146.91989	145.27686	144.85217	144.58154	144.38002
12	260.94135	174.51744	151.92743	146.43491	145.15779	144.8269	144.58086	144.38
12.5	249.37521	170.11682	150.55795	146.07118	145.07446	144.81049	144.58047	144.38
13	238.96568	166.37631	149.46236	145.79839	145.01613	144.79982	144.58026	144.38
13.5	229.59712	163.19687	148.58589	145.5938	144.97528	144.79288	144.58014	144.38
14	221.16542	160.49434	147.8847	145.44034	144.9467	144.78838	144.58008	144.38
14.5	213.57687	158.19719	147.32376	145.32526	144.9267	144.78545	144.58005	144.38
15	206.74718	156.24461	146.87502	145.23894	144.91269	144.78354	144.58003	144.38
15.5	200.60046	154.58492	146.51601	145.17421	144.90288	144.7823	144.58002	144.38
16	195.06842	153.17418	146.22881	145.12566	144.89603	144.78149	144.58	144.38
16.5	190.08957	151.97505	145.99905	145.08923	144.89122	144.78098	144.58	144.38
17	185.60861	150.9558	145.81525	145.06192	144.88785	144.78062	144.58	144.38
Base value	600.297	600.297	600.297	600.297	600.297	600.297	600.297	600.297
PV (UserS)	-0.00067	212.47711	330.67072	394.08387	426.75356	442.83617	453.70476	455.70261
PV(%)	0.00%	35.40%	55.08%	65.65%	71.09%	73.77%	75.58%	75.91%

Table C-2: Cost Impact Results

Case Study 2 - Cost Variance Due to Change in Development Rate								
Time (Month)	**200 u/s**	**260 u/s**	**320 u/s**	**380 u/s**	**440 u/s**	**540 u/s**	**650 u/s**	**850 u/s**
0	179226.0313	179226.031	179226.031	179226.031	179226.031	179226.031	179226.031	179226.031
0.5	359334.0313	449379.031	539424	629469	719514	809559	989649	1169739.13
1	521442.0313	679020.75	827595	967164.75	1097730	1219290.75	1435398.75	1615488.75
1.5	667355.4375	874233.938	1058150.75	1220456.88	1362502.75	1485639.38	1680586.75	1816104.5
2	798701.875	1040191.25	1242623.63	1410456.38	1547876.25	1658800.5	1815478.75	1906424.25
2.5	916950	1181294.63	1390244.5	1553001.5	1677686.25	1771406.75	1889727	1947131.88
3	1023428.125	1301291.75	1508405.13	1659978.75	1768626.13	1844678.38	1930650.25	1965546
3.5	1119290.5	1403328.13	1602979.25	1740264.25	1832343.5	1892371.13	1953237.88	1973926
4	1205504.375	1490004.75	1678592	1800439.5	1876914.63	1923348.13	1965653.25	1977704.63
4.5	1283066	1563652.63	1739058.88	1845551.63	1908098.75	1943471.5	1972477.88	1979408.88
5	1352856	1626239.88	1787420.75	1879376	1929920	1956546	1976229.38	1980177.75
5.5	**1415659.125**	**1679432.25**	**1826104.5**	**1904739.38**	**1945191**	**1965041.38**	**1978291.75**	**1980524.63**
6	1472178	1724642.38	1857048.63	1923759.5	1955878.75	1970562	1979425.63	1980681.38
6.5	1523043.25	1763069.25	1881802.5	1938023.5	1963359.25	1974149.75	1980049	1980752
7	1568820.875	1795731.25	1901604.75	1948720.75	1968595	1976481.25	1980391.75	1980784
7.5	1610020.375	1823493.5	1917446.25	1956743.5	1972259.88	1977996.63	1980580.13	1980798.38
8	1647099.5	1847091.25	1930119.25	1962760.25	1974825.25	1978981.5	1980683.75	1980805
8.5	1680470.875	1867149.25	1940257.63	1967273	1976620.75	1979621.75	1980740.75	1980808
9	1710504.75	1884198.5	1948368.13	1970657.25	1977877.63	1980037.75	1980772	1980809.25
9.5	1737535.5	1898690.38	1954856.63	1973195.63	1978757.5	1980308.25	1980789.38	1980809.88
10	1761863	1911008.38	1960047.38	1975099.38	1979373.38	1980484	1980798.88	1980810.13
10.5	1783757.75	1921478.63	1964200	1976527.13	1979804.5	1980598.25	1980804	1980810.25
11	1803463	1930378.38	1967522.13	1977597.88	1980106.25	1980672.5	1980806.88	1980810.38
11.5	1821197.75	1937943.25	1970179.75	1978401	1980317.5	1980720.75	1980808.5	1980810.38
12	1837159	1944373.25	1972305.88	1979003.38	1980465.38	1980752.13	1980809.38	1980810.38
12.5	1851524.125	1949838.88	1974006.75	1979455.13	1980568.88	1980772.5	1980809.75	1980810.38
13	1864452.75	1954484.63	1975367.5	1979793.88	1980641.38	1980785.75	1980810	1980810.38
13.5	1876088.5	1958433.5	1976456.13	1980048	1980692	1980794.38	1980810.25	1980810.38
14	1886560.625	1961790	1977326.88	1980238.63	1980727.5	1980800	1980810.25	1980810.38
14.5	1895985.625	1964643	1978023.63	1980381.63	1980752.38	1980803.63	1980810.38	1980810.38
15	1904468.125	1967068.13	1978581	1980488.75	1980769.75	1980806	1980810.38	1980810.38
15.5	1912102.375	1969129.5	1979026.88	1980569.13	1980782	1980807.5	1980810.38	1980810.38
16	1918973.125	1970881.63	1979383.5	1980629.5	1980790.5	1980808.5	1980810.38	1980810.38
16.5	1925156.875	1972370.88	1979668.88	1980674.75	1980796.5	1980809.13	1980810.38	1980810.38
17	1930722.25	1973636.88	1979897.25	1980708.63	1980800.63	1980809.63	1980810.38	1980810.38
Base cost	1415659	1415659	1415659	1415659	1415659	1415659	1415659	1415659
Cost diff		$263,773.25	$410,445.50	$489,080.38	$529,532.00	$549,382.38	$562,632.75	$564,865.63
CV		18.63%	28.99%	34.55%	37.41%	38.81%	39.74%	39.90%

Appendix D – Case Study 3 Results

Table D-1: Performance Reduction Impact Results

Case Study 3 - Performance Reduction Impact to Schedule								
Time (Month)	**Per100%**	**Per95%**	**Per90%**	**Per 85%**	**Per80%**	**Per75%**	**Per70%**	**Per65%**
0	5944.729	5944.74365	5944.75879	5944.77393	5944.78857	5944.80371	5944.81885	5944.83398
0.5	5364.74365	5408.2583	5451.77344	5495.28857	5538.80322	5582.31836	5625.8335	5669.34863
1	4842.76563	4921.40625	5000.7002	5080.64697	5161.24512	5242.49658	5324.40039	5406.95703
1.5	4372.99805	4479.60059	4587.98047	4698.15186	4810.12793	4923.92529	5039.55762	5157.04004
2	3950.22681	4078.68213	4210.36182	4345.31982	4483.6084	4625.28369	4770.3999	4919.0127
2.5	3569.76221	3714.8772	3864.8689	4019.85986	4179.97266	4345.33447	4516.07275	4692.31787
3	3227.38818	3384.76758	3548.78589	3719.66577	3897.63354	4082.92334	4275.77441	4476.43115
3.5	2919.2771	3085.21802	3259.59399	3442.75977	3635.08032	3836.93457	4048.71338	4270.81982
4	2641.92749	2813.32617	2994.93164	3187.26147	3390.85229	3606.26563	3834.08521	4074.91797
4.5	2392.28784	2566.55884	2752.73975	2951.53784	3163.69336	3389.98608	3631.23364	3888.29346
5	2167.59961	2342.60474	2531.12109	2734.06958	2952.42212	3187.21021	3439.52515	3710.51953
5.5	1965.37378	2139.35986	2328.3335	2533.44849	2755.93311	2997.10107	3258.35352	3541.18262
6	1783.36755	1954.91199	2142.77979	2348.37231	2573.19507	2818.87036	3087.14282	3379.88574
6.5	1619.5603	1787.52405	1972.99634	2177.6377	2403.24683	2651.77734	2925.34692	3226.24878
7	1472.13306	1635.61865	1817.6438	2020.13416	2245.19409	2495.12671	2772.44897	3079.90869
7.5	1339.44812	1497.76416	1675.49585	1874.83691	2098.20459	2348.26636	2627.95996	2940.51929
8	1220.03149	1372.66101	1545.4303	1740.79993	1961.50427	2210.58472	2491.41772	2807.75049
8.5	1112.5564	1259.12976	1426.42017	1617.15076	1834.37292	2081.50781	2362.38501	2681.28809
9	1015.82892	1156.1001	1317.52588	1503.08435	1716.14075	1960.49854	2240.44922	2560.83276
9.5	928.77399	1062.60071	1217.88757	1397.85803	1606.18469	1847.05212	2125.21997	2446.09912
10	850.42462	977.75006	1126.71851	1300.78674	1503.92554	1740.69604	2016.32825	2336.81519
10.5	779.9101	900.74799	1043.29871	1211.2384	1408.82458	1640.98718	1913.42554	2232.72217
11	716.44714	830.86859	966.96973	1128.63013	1320.38062	1547.51013	1816.1825	2133.57349
11.5	659.33044	767.45306	897.12866	1052.42395	1238.12781	1459.87549	1724.28784	2039.13464
12	607.92542	709.90344	833.22406	982.12384	1161.63269	1377.71802	1637.44739	1949.18152
12.5	561.66089	657.67719	774.75134	917.27191	1090.49219	1300.69531	1555.38318	1863.50122
13	520.02277	610.2818	721.24884	857.44604	1024.33154	1228.48657	1477.8324	1781.89063
13.5	482.54852	567.27057	672.29401	802.25665	962.80219	1160.79089	1404.547	1704.15662
14	448.82166	528.23779	627.50037	751.34442	905.57983	1097.32617	1335.29224	1630.11499
14.5	418.46753	492.81558	586.51416	704.37793	852.36304	1037.828	1269.84656	1559.59033
15	391.14874	460.66989	549.01184	661.05133	802.8714	982.04846	1208.00037	1492.41553
15.5	366.56189	431.49771	514.69714	621.08252	756.84418	929.75513	1149.55566	1428.43152
16	344.43369	405.02396	483.29929	584.2113	714.03888	880.7301	1094.32544	1367.48682
16.5	324.51831	380.99905	454.57016	550.19763	674.22998	834.76917	1042.13293	1309.43701
17	306.59448	359.19641	428.28305	518.82001	637.20764	791.68079	992.81091	1254.14453
17.5	290.46301	339.41055	404.23038	489.87415	602.77692	751.2854	946.20166	1201.47839
18	275.94473	321.45486	382.22214	463.17157	570.75635	713.41473	902.15588	1151.31396
18.5	262.87827	305.16006	362.08459	438.53845	540.97717	677.91101	860.53265	1103.53235
19	251.11842	290.37253	343.65875	415.81439	513.28259	644.62628	821.19873	1058.02039
19.5	240.53458	276.95285	326.7991	394.8515	487.52661	613.42181	784.02814	1014.67017
20	231.00911	264.77451	311.37253	375.51318	463.57355	584.1676	748.90192	973.37915
20.5	222.4362	253.72266	297.25723	357.67358	441.29718	556.74182	715.70764	934.04944
21	214.72058	243.6931	284.34171	341.21658	420.5802	531.03015	684.33905	896.58789
21.5	207.77652	234.59128	272.52402	326.03497	401.31335	506.92545	654.69574	860.90576
22	**201.52687**	**226.33138**	**261.71085**	**312.02997**	**383.39523**	**484.3273**	**626.6828**	**826.91852**
22.5	195.90218	218.83551	251.81674	299.11035	366.73138	463.14151	600.21063	794.54565
23	190.83997	212.03302	242.76367	287.19196	351.23398	443.27988	575.1944	763.71051
23.5	186.28397	205.85974	234.48012	276.19727	336.82141	424.65958	551.55402	734.34003
24	182.18358	200.25751	226.90065	266.05469	323.41769	407.20303	529.21393	706.36469
24.5	178.49321	195.17348	219.96544	256.69812	310.95227	390.83752	508.10251	679.71814
25	175.17189	190.55972	213.61974	248.06671	299.35941	375.49487	488.15225	654.33728
Base value	201.5269	201.5269	201.5269	201.5269	201.5269	201.5269	201.5269	201.5269
PV	3E-05	-24.80448	-60.18395	-110.50307	-181.86833	-282.8004	-425.1559	-625.39162
PV(%)	0.00%	-0.43%	-1.04%	-1.91%	-3.14%	-4.88%	-7.33%	-10.78%

Appendix E – Case Study 4 Results

Table E-1: Development Rate Impact Results

Case Study 4 - Development Rate Impact				
Time (Month)	200 u/s	260 u/s	130 u/s	100 u/s
0	5799.7002	5799.6001	5799.7998	5799.8501
0.5	4958.7002	4668.6001	5248.7998	5393.8501
1	4239.64502	3758.14502	4750.14453	5016.27002
1.5	3624.85278	3025.22876	4298.86182	4665.12061
2	3099.20557	2435.2312	3890.45117	4338.55176
2.5	2649.7771	1960.28308	3520.83936	4034.84253
3	2265.51587	1577.94983	3186.34058	3752.39307
3.5	1936.97253	1270.17163	2883.61914	3489.71509
4	1656.06799	1022.41016	2609.65625	3245.42456
4.5	1415.89465	822.96216	2361.71997	3018.23438
5	1210.54639	662.40649	2137.33765	2806.94751
5.5	1034.97363	533.15918	1934.27161	2610.45068
6	884.85901	429.11514	1750.49683	2427.70874
6.5	756.51093	345.35968	1584.18066	2257.75854
7	646.77332	277.93655	1433.66455	2099.70483
7.5	552.94769	223.66092	1297.44739	1952.71484
8	472.72678	179.96904	1174.1709	1816.01428
8.5	404.13788	144.79707	1062.60571	1688.88281
9	345.49438	116.48363	961.6391	1570.65051
9.5	295.35419	93.69132	870.2644	1460.69446
10	252.48431	75.34351	787.57031	1358.4353
10.5	215.83058	60.57353	712.73212	1263.33435
11	184.49164	48.68369	645.00354	1174.8905
11.5	157.69685	39.11237	583.70923	1092.6377
12	134.78731	31.40746	528.23785	1016.14258
12.5	**115.19965**	**25.205**	**478.03625**	**945.00208**
13	98.45219	20.21203	432.60382	878.84143
13.5	84.13313	16.19268	391.48746	817.31201
14	71.89032	12.95711	354.27716	760.08966
14.5	61.42273	10.35247	320.60184	706.87286
15	52.47293	8.25574	290.12567	657.38129
15.5	44.82085	6.56787	262.54474	611.35406
16	38.27833	5.20914	237.584	568.54877
16.5	32.68447	4.11536	214.99452	528.73987
17	27.90173	3.23486	194.55104	491.71762
17.5	23.81248	2.52606	176.0497	457.2869
18	20.31617	1.95548	159.30598	425.26633
18.5	17.32682	1.49616	144.15291	395.48718
19	14.77093	1.12641	130.43938	367.79257
19.5	12.58565	0.82876	118.02864	342.03659
20	10.71723	0.58915	106.79692	318.08353
20.5	9.11973	0.39627	96.63222	295.80719
21	7.75387	0.241	87.43316	275.09018
21.5	6.58606	0.116	79.10801	255.82336
22	5.58758	0.01538	71.57375	237.90523
22.5	4.73388	-0.06562	64.75524	221.24136
23	4.00397	-0.13082	58.58449	205.74397
23.5	3.37989	-0.18331	52.99997	191.33139
24	2.84631	-0.22557	47.94597	177.92769
24.5	2.39009	-0.25958	43.3721	165.46225
25	2.00003	-0.28696	39.23275	153.86938
Base value	115.1997	115.1997	115.1997	115.1997
PF	-5E-05	-89.9947	362.83655	829.80238
PF(%)	0.00%	-1.55%	6.26%	14.31%

Case Study 4 - Cost Factor with Performance Adjustment				
Time (Month)	200 u/s	260 u/s	130 u/s	100 u/s
0	5544	5544	5544	5544
0.5	2789540	3749538	1829540	1349542
1	5169857	6763454	3480259	2599460
1.5	7205027	9189656	4974159	3761884
2	8945098	11142748	6326138	4842938
2.5	10432859	12714988	7549678	5848319
3	11704894	13980641	8656983	6783323
3.5	12792485	14999492	9659094	7652876
4	13722374	15819666	10566004	8461561
4.5	14517430	16479907	11386758	9213638
5	15197202	17011400	12129540	9913069
5.5	15778408	17439254	12801758	10563540
6	16275339	17783674	13410114	11168478
6.5	16700214	18060934	13960678	11731071
7	17063484	18284128	14458937	12254282
7.5	17374078	18463798	14909862	12740869
8	17639636	18608434	15317949	13193394
8.5	17866690	18724864	15687268	13614243
9	18060820	18818592	16021502	14005632
9.5	18226800	18894042	16323983	14369624
10	18368714	18954780	16597728	14708136
10.5	18490050	19003672	16845468	15022953
11	18593792	19043032	17069672	15315732
11.5	18682492	19074716	17272578	15588017
12	18758332	19100222	17456208	15841242
12.5	**18823172**	**19120754**	**17622392**	**16076741**
13	18878612	19137284	17772788	16295755
13.5	18926014	19150588	17908898	16499439
14	18966542	19161300	18032076	16688864
14.5	19001192	19169922	18143552	16865030
15	19030820	19176862	18244440	17028864
15.5	19056150	19182450	18335742	17181230
16	19077808	19186948	18418370	17322930
16.5	19096326	19190568	18493148	17454710
17	19112158	19193484	18560824	17577266
17.5	19125696	19195830	18622070	17691244
18	19137270	19197718	18677496	17797244
18.5	19147166	19199240	18727658	17895822
19	19155626	19200464	18773056	17987500
19.5	19162860	19201448	18814140	18072762
20	19169046	19202242	18851320	18152054
20.5	19174334	19202880	18884968	18225798
21	19178854	19203394	18915420	18294378
21.5	19182720	19203808	18942980	18358158
22	19186026	19204140	18967920	18417472
22.5	19188852	19204410	18990492	18472636
23	19191268	19204624	19010920	18523936
23.5	19193334	19204798	19029406	18571648
24	19195100	19204938	19046136	18616018
24.5	19196610	19205052	19061278	18657284
25	19197902	19205142	19074980	18695660
Base value	18823172	18823172	18823172	18823172
CV	0	-297582	1200780	2746431
CV(%)	0.00%	-1.55%	6.25%	14.30%