DISSERTATION

# EVALUATION OF A MODEL-BASED APPROACH TO ACCREDITING UNITED STATES GOVERNMENT INFORMATION TECHNOLOGY SYSTEMS FOLLOWING THE AUTHORIZATION TO OPERATE PROCESS

Submitted by

Edan Christopher Sanchez

Department of Systems Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2025

Doctoral Committee

Advisor: Thomas H. Bradley

John M “Mike” Borky
Ron Sega
Jianguo Zhao

ABSTRACT

EVALUATION OF A MODEL-BASED APPROACH TO ACCREDITING UNITED STATES GOVERNMENT INFORMATION TECHNOLOGY SYSTEMS FOLLOWING THE AUTHORIZATION TO OPERATE PROCESS

This research project explores Model-Based Systems Engineering (MBSE) methodology as a modernized, alternative strategy to improve the United States Government's (USG) accreditation processes and procedures for accepting new/updated information systems. While the primary goal is to significantly accelerate the transition of advanced technology to operational environments, it is imperative that we take advantage of the potential benefits realized through the implementation of a model-based process.

While this dissertation primarily focuses on defense systems within the USG domain, the principles discussed are applicable in a broader context. This research focuses on the application of MBSE to defense Information Technology (IT) systems, or simply Information Systems (IS) that requires an Authorization to Operate (ATO)[1]. Currently, the security accreditation process for obtaining an ATO for Government systems is primarily document-centric. This approach often leads to frequent schedule overruns, significantly increasing costs and negatively impacting stakeholders. This issue is particularly pronounced for large, software- and data-intensive systems, such as those utilized by the Department of Defense (DoD), Intelligence, and command and control (C2) operations. The complexity of authorization is significantly magnified when systems

[1] Also known or referred to as Authority to Operate

incorporate third-party applications requiring independent accreditation, creating cascading dependencies that impact overall system security and deployment timelines, as well as for real-time systems that must meet stringent cybersecurity requirements while adhering to strict process deadlines. Mission effectiveness is compromised when operators and end users experience delays in accessing essential tools. The trend toward implementing these types of IT systems is accelerating, highlighting the urgent need to enhance their authorization processes.

The proposed approach aims to capture the existing ATO process using a formal Systems Modeling Language (SysML) model. This model will facilitate an analysis to identify bottlenecks, redundant activities, missing interfaces, and other areas of concern. Once the model is developed and analyzed, corrective actions and proposed improvements will be introduced to enhance the process model. The potential benefits will be quantified in terms of speed-to-operations, particularly regarding schedules, as well as improvements in consistency and efficiency throughout the end-to-end process, ultimately leading to a potential reduction in overall system costs. Furthermore, the anticipated gains will be validated through modeling and analysis of the enhanced process as applied to a representative IT system, also represented in SysML. This modeled IT system will reflect the cloud-centric environments currently found in operational contexts, utilizing approved tools and technologies available to development contractors.

This research will assess the impact of MBSE on the ATO. It aims to measure MBSE's effectiveness in mitigating inconsistencies, streamlining system deployment timelines, enhancing quality, reducing costs, and delivering other advantages in this practical context. The conclusions drawn from this study will establish a framework for investing in the modernization of the ATO towards a systems-engineered, model-based approach, particularly within the realm of USG systems development. The model-based ATO process will facilitate integration with the federal

Digital Engineering (DE) transformation as DE continues to broaden its presence within the federal systems engineering landscape.

# ACKNOWLEDGEMENTS

This achievement has been a remarkable journey, and there are a few incredible people who have made it all possible. I am thrilled to dedicate this dissertation to them:

To my wonderful wife, Jayme Marie Sanchez—this journey would not have been possible without your unwavering friendship, love, and encouragement. Your endless support has fueled my passion and nurtured my confidence every step of the way. I am deeply grateful for all you do and always believing in me. Thank you.

To my incredible parents, Moses "Jake" and Lynn Sanchez, thank you for encouraging my dreams and instilling in me the belief that I could achieve anything. To my brother, Jeff Sanchez, and his wonderful wife, Gina, along with their precious kids, Jeremiah and Delilah, your constant love and support have been a source of strength for me throughout this journey. I am genuinely grateful to have you all by my side.

To my beautiful children, Riley Jack, Sydney Paige, and Stone Alexander, your joy, curiosity, and affection have inspired me every step of the way. You motivate me to strive for excellence and set the bar high for myself. And to my stepchildren, Mayzie Jane and Emery Jameson Lundy, thank you for being part of this adventure.

I am also incredibly grateful to the remarkable professors and educators at Colorado State University (CSU), who have empowered me with the knowledge needed to reach this milestone. A special thank you to Dr. John M "Mike" Borky for believing in me and championing my research topic. I appreciate my committee members' invaluable time and feedback, as well as their stimulating conversations and encouragement to keep me on track. And, of course, a heartfelt

thanks to Dr. Thomas Bradley for his steady support and insightful feedback that guided me through to the finish line.

Lastly, I want to thank Mr. Warren Ayr for graciously participating and dedicating his time to the interview portion of this dissertation.

I am filled with appreciation for each of you who contributed to this journey. Thank you for helping me make this dream a reality.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

# CHAPTER 1 – INTRODUCTION

## 1.1 BACKGROUND

The following paragraphs provide a brief history and background information about the ATO process and its effectiveness in securing and accrediting USG ISs assets to operate in a forward-facing/production environment[2]. This chapter also includes an interview with a Subject Matter Expert (SME), an outline of the dissertation, a problem synopsis, a literature review, and a proposed solution.

### 1.1.1 TRACING THE EVOLUTION OF THE ATO PROCESS

In 1972, the DoD published DoD Directive 5200.28, Security Requirements for Automated Information Systems, which it later updated in 1988. Along with DoD Directive 5200.28-STD, also known as the "Orange Book," which was released in 1983, these directives formed the basis for the testing and accreditation of systems within DoD. These directives left room for interpretation about the process, which resulted in each of the military services specifying in its regulations similar but separate two accreditation processes to be used within that service. From the beginning, the accreditation processes used within the DoD focused on discrete, individual systems as the target of testing and accreditation [1].

In 1997, the DoD published DoD Instruction 5200.40, DoD Information Technology Security Certification and Accreditation Process (DITSCAP). The objective of this new regulation was "to establish a DoD standard infrastructure-centric approach that protects and secures the

[2]. The term "operations" will also be utilized throughout this dissertation to convey the production/operational environment effectively.

entities comprising the Defense Information Infrastructure (DII). The set of activities presented in the DITSCAP standardized the C&A process for single Information Technology (IT) entities, leading to more secure system operations and DII. The process considers the system mission, environment, and architecture while assessing the impact of the operation of that system on the DII [2]." This effort to synchronize the certification and accreditation process across the entire DoD and begin assessing risks in terms of the enterprise was a step in the right direction. However, even DITSCAP still focused on discrete, individual systems as the target of testing and accreditation.

In 2006, the DoD replaced DITSCAP with the DoD Information Assurance Certification and Accreditation Process (DIACAP) as published interim guidance, and later (2007) finalized in DoD Instruction 8510.01. Ostensibly, this change was made to "address the paradigm shift in IA security from an individual information system-level approach to a DoD-wide enterprise approach of securing information systems in a net-centric environment and for supporting the implementation of IA security during a system's life cycle [2]." This sounds a lot like what the DITSCAP was intended to do, but the DIACAP still tested and accredited individual systems and discrete enclaves (e.g., local area networks) that could be tested and accredited as one "system."

In alignment with broader federal government initiatives, the DoD has adopted the National Institute of Standards and Technology (NIST) Risk Management Framework (RMF) as specified in NIST Special Publication (SP) 800-37, and the NIST-developed set of controls published in NIST SP 800-53. The current DIACAP aligns closely with the intent of the process called out in the NIST RMF. The most significant changes the DoD will have to adjust to will be the new RMF-related language (e.g., "Authorizing Official" under the RMF versus "Designated Approving

Authority" under the DIACAP) and, more significantly, the latest set of controls in NIST SP 800-53 [2].

### 1.1.2 HISTORICAL EFFECTIVENESS OF THE ACCREDITATION PROCESS

Going back thirteen years from the authorship of this document, the DoD was already struggling to field operational systems securely. The DoD Strategy for Operating in Cyberspace, published in July 2011, ups the ante on the Certification and Accreditation (C&A) processes. Strategic Initiative Five in the document calls upon the DoD's acquisition process for information technology to become more dynamic and agile. It will do this by adopting five principles:

- Speedier fielding processes,
- Employing incremental development and testing,
- Sacrificing or deferring customization when possible,
- Applying differing levels of oversight based on the prioritization of critical systems,
- Focusing on the security of the systems that DoD buys, including software and hardware.

The strategy calls for the rapid movement of concepts from an innovative idea to a pilot program to scaled adoption across the DoD enterprise [2].

"For many years, the Department of Defense (DoD) has used very formalized processes for authorizing the operation of its information systems. This authorization process, known as accreditation within the DoD, has always been based on certification testing of those systems and assessing the risks associated with operating those systems on the DoD's Global Information Grid (GIG). Despite using these various costly and process-intensive methods for certification and accreditation (C&A), it is questionable whether these processes have improved the security of DoD systems and networks commensurate with the cost and effort involved. Further, given current

advances in systems security technologies, recent changes in DoD's strategy for operating in cyberspace, and even the very structure of the DoD's enterprise networks in the near future, should (or even can) the DoD continue to test and authorize information systems using these same methodologies [2]?" This dissertation explores innovative approaches to modernizing methodologies by leveraging the best practices of Digital Engineering (DE), with a particular focus on MBSE.

### 1.1.3 INTERVIEW WITH SYSTEM SECURITY PROFESSIONAL REGARDING THE ATO

In a recent interview with cybersecurity expert Warren Ary, a Certified Information Systems Security Professional (CISSP) who helps complex systems achieve ATO, he discussed his challenges with the current ATO process.

What are some of the challenges you see with the current process?

> *A challenge I see is the lack of acceptance of previously approved items. For example, Amazon Web Services (AWS) is a highly utilized cloud asset. AWS cloud services have been given an accreditation and are constantly being reevaluated. Unfortunately, some people believe that even if you use a pre-approved service the asset still requires an evaluation before being used in testing and/or operations. For example, I wanted to utilize a pre-approved AWS machine image of Windows 2019 server in my system, I must have an assessor evaluate it to ensure its security posture. By virtue of being pre-approved, this means it's already been hardened and scanned/patched for vulnerabilities. Having someone evaluating the pre-approved asset just adds more duplication to the process.*

Are there opportunities to be more efficient by using auto-generated documents from templates? If so, what documents?

*A key challenge I see in the current process is the amount of reliance of human intervention versus automation. Some companies still rely on manual development of security documentation required to be submitted in the security package for a system accreditation process. There are several different automation tools available that would generate the same required documents in half the time it does to manually generate the same documents.*

Where do you see the most redundancies?

*The most redundancy I've encountered was dealing with getting an application through the Authorization to Operate (ATO) process.*

*The redundancy occurs due to the requirement to generate and/or submit an entire separate security package that could contain upwards to 15+ artifacts (security name for documentation) which includes such items as System Security Plan (SSP), system drawings, access controls, etc. Plus, the application requires a separate Assessment and Authorization process to obtain an ATO.*

*Ideally, the application is deployed on a system with an approved ATO; for example: new application is installed on an additional Linux server within a system. Since the additional server is a "Like & Kind" infrastructure asset, the installment of the application should just inherit the ATO status. The only required update is adding the application as an appendix to the master SSP; the appendix would identify key security items that pertain specifically to the application.*

In your experience, how long is a typical ATO process. Where do you think most of the cost is incurred?

*From my experience, the ATO process usually takes between 6-8 months. The variances are due to the numerous steps and personnel involved in the ATO process.*

*Typical ATO process involves the following steps:*

*Step 1 (Registration): a program registers the system in some sort of "System of Record" tool; XACTA 360 is the most used tool*

*Step 2 (Categorize) and 3 (Select): program identifies the importance of the system, which then generates a list of security controls needed to be applied towards the system*

*Step 3 (Implement): once the list of security controls is identified, program/developers/administrators go to work on applying them to the system. This step may also require hardening of the system to satisfy certain controls. I believe this is where the bulk of the costs are incurred during the ATO process, which encounters such items as personnel costs, equipment (physical and/or cloud), etc.*

*Step 4 (Assessment): after the controls have been implemented/complied with, assessors are coordinated with to conduct an evaluation of the system; normal system evaluation takes about 30 days, but I've seen it take longer. Results of the evaluation depict the timeline for this step, if there are numerous findings then the evaluation is temporary halted to allow for resolution.*

*Step 5 (Authorization): This is typically a risk acceptance decision; basically, it means how susceptible the system is to the exploitation of any vulnerabilities. This is the step where the ATO is granted for the system. The ATO ranges from 1 to 3 years. Usually, within 6 months of the ATO expiration, the system security package is revamped/edited in preparation for reaccreditation.*

As highlighted by the response from a system security specialist (CISSP), there are significant challenges in the current process for system accreditation. These challenges arise from various factors that contribute to delays and hinder effective progress.

## 1.2 CONTENT OF THE DISSERTATION

This dissertation presents a contemporary approach to accrediting U.S. Government Information Systems (IS). By streamlining a cumbersome, document-heavy process that has been in place for over 50 years, this research introduces a model for accreditation based on the RMF. This model aims to enhance the accreditation process by minimizing documentation, and more effectively identify and resolve inconsistencies in order to encourage the reuse of system artifacts, promote collaboration, and centralize data.

Chapter One (1) introduces the research and its methodologies while providing essential background information. It offers a historical perspective to contextualize the needs and methods previously employed for securing and accrediting systems. An interview with a current government contractor holding a Certified Information Systems Security Professional (CISSP) credential is included to establish the foundation for the current ATO processes. Additionally, the chapter outlines the content of the dissertation, discusses the motivation behind the research, and includes a literature review. The chapter concludes with a proposed solution that is supported by the subsequent chapters of the dissertation.

Chapter Two (2) describes the specific implementation of the research objectives and the evaluation process through a series of questions and related tasks. The research methodology comprises three primary questions intended to guide a set of tasks designed to conduct activities and analyses that assess the impact of applying MBSE to the ATO.

Chapter Three (3) explores the motivation for modeling the ATO and presents an example system, referred to as the "Asset Management System," which is representative of the current technologies approved and utilized by the U.S. Government. This chapter also describes the modeling tools and language employed to develop the models for experimentation and analysis.

Chapter four (4) describes the assessment of research question one (1), which was introduced in chapter two (2), and the effect of MBSE regarding inconsistencies. An examination of how MBSE can address inconsistencies across requirements, documentation, and traceability is explored.

Chapter five (5) describes the assessment of research question two (2), which was introduced in chapter two (2), and the effect of MBSE regarding reusability. It investigates how MBSE can be used to promote and enable reuse and measures it against a discouraged copy/paste process and an encouraged object-oriented paradigm. Examples are provided of what the industry considers proper and improper reuse practices.

Chapter six (6) assesses research question three (3), which was introduced in chapter two (2), focusing on the effects of MBSE and its potential for reducing costs and saving time by implementing an MBSE ATO.

Chapter seven (7) provides a comprehensive summary of the dissertation, synthesizing the results, drawing conclusions, and offering recommendations for future research. The synthesis of results encapsulates the findings related to the research questions addressed in chapters 4, 5, and 6. From this synthesis, conclusions are formulated, along with suggestions for areas of future investigation.

## 1.3 PROBLEM SYNOPSIS

Accreditation of USG ISs is essential to ensure their functionality and security before they are deployed in the operational environment. This accreditation process is formally known as the ATO and relies heavily on documentation. Currently, the security accreditation process for government systems seeking ATO has resulted in significant schedule overruns, impacting project stakeholders

and taxpayers. This issue is particularly pronounced in large, software- and data-intensive systems used by Intelligence and Command and Control communities.

Due to these schedule delays, end users often do not receive the necessary tools for their job functions in a timely manner, which can impact the mission. Furthermore, accreditation challenges are intensified in many systems that incorporate third-party providers' application software and other components. These providers are usually not directly controlled by the organization responsible for the system and typically undergo their own independent security testing and accreditation. This process has driven many product owners and developers to find workarounds to bypass some of the daunting and time-consuming processes associated with accreditation. They must rely instead on their legacy tools and techniques, which also increases the cost of missed project deliveries and may require an extension of the legacy software accreditation. The situation will likely worsen as Government policy mandates a transition from legacy accreditation processes to the RMF, which typically involves a significantly larger number of security controls, all of which must be tested and verified. The time spent brainstorming and determining ways to achieve accreditation could be better spent enhancing the product for delivery.

Security is often incorporated as the final step in the system development and acquisition process. Due to resource limitations, the review process for assessing the system's security posture can take 45 days or more. If an issue is identified, if a document or other deliverable is missing, or if clarification is needed, the delivery deadline may already have passed.

Several root causes contribute to the ongoing challenges associated with the security accreditation of software-intensive systems. Foremost among these is the failure to incorporate technically skilled security professionals within the project teams and insufficient planning for accreditation timelines in the development schedule. Additional factors include a shortage of

experienced government program managers, a misunderstanding of the complexity involved, and inadequate resourcing for accreditation activities. The overarching issue remains the failure to recognize the security accreditation process as a distinct yet essential parallel “life cycle” that is critical for achieving initial operational capability and for the ongoing sustainment of operational systems.

A compelling need exists to streamline the accreditation process while ensuring robust security measures are in place. A model-based approach will help identify areas that significantly impact the schedule. Analyzing lessons learned from previous delays will lead to innovative strategies for accreditation, along with developing new workflows and tools tailored to system requirements. Rather than employing a one-size-fits-all method, the accreditation process can be customized to address the system's specific needs or application under development, with security requirements identified from the project's inception. This dissertation explores various opportunities to leverage MBSE to enhance the system accreditation process.

System and network administrators generally have too much to do and deal with many competing priorities. If keeping the system/network secure is not at the top of their priorities, administrators may put off their security-related duties (e.g., patching and proper configuration management) until a certification test is imminent. Then, they will “surge” to clean up the security posture of their system/network just for the test. This has been a common scenario identified by US Army Information Systems Engineering Command (USAISEC) certification testers. The result of this type of paradigm is systems and networks that remain in a poor state of security until just before a certification test takes place [2].

- A formal architecture model allows capture, maintenance, and visualization of the configuration baseline and processes or behaviors of a system or system-of-systems.

Artifacts exported from such a model can be focused on specific aspects, such as security features and functions. They can effectively provide the information security testers, analysts, and accreditation authorities need. For example, architecture data can support effective vulnerability assessments and penetration testing planning.

- An architecture model allows rigorous and auditable flow-down of requirements, including security requirements, to components, interfaces, and processes to ensure all requirements have been accounted for in system design and to provide a basis for requirements verification and validation.
- As cloud and service-oriented architecture (SOA) becomes the preferred approach to implementing large information systems and enterprises, the power of architecture modeling in defining services and, especially, service interfaces are invaluable in system accreditation involving security services, both for internal services and for those which a system provides or consumes as a participant in a larger enterprise.
- An architecture model provides a basis for configuration, data, and change management. For example, a model captures dependencies among system elements to support defining required regression testing following a design change. The model also serves as a searchable repository of system and component information, including functions such as archiving data from previous security testing to keep subsequent testing to a feasible minimum.
- Modern system architecture modeling tools facilitate the development of behavioral diagrams to visualize the processes they represent. This "executable architecture" technique is valuable for assessing the operation of security controls.

As technology advances rapidly, the traditional ATO process struggles to keep up. This discrepancy allows adversaries to deploy sophisticated systems and technology in operational environments more quickly than what is currently acceptable under the U.S. Government's existing practices and protocols. The enduring challenge of obtaining ATO has often resulted in project delays and cost overruns. However, modernization strategies, such as MBSE, are now being promoted as effective methods to expedite and streamline the development and deployment process. Additionally, agile development processes have enhanced the deployment of solutions to end-users. The ATO should not act as a barrier to the swift deployment of solutions; instead, it should seamlessly integrate into the development process, providing risk assessments and management from the outset. This dissertation explores the opportunity to streamline the ATO by adopting MBSE processes, tools, and techniques.

## 1.4 LITERATURE REVIEW

The following literature review was conducted to explore potential solutions to the challenges outlined in section 1.3, Problem Synopsis. The initial step involved investigating and documenting the existing limitations and challenges of the traditional Document Based – ATO (DB-ATO) process. Subsequently, the necessary process steps to achieve ATO under the RMF were examined. Finally, a comparative analysis of current systems engineering practices, such as the DB-ATO, was performed alongside more contemporary MBSE processes. This investigation ultimately informed the proposed solution presented in section 1.5.

### 1.4.1 LIMITATIONS OF BASELINE (DOCUMENT-CENTRIC) SYSTEMS ENGINEERING

In the traditional Systems Engineering approach, the lifecycle of a system produces a substantial number of documents. For large, complex systems such as aircraft, satellites, and power

plants, this documentation encompasses a wide range of materials, including the Requirements Specifications, Requirement Traceability Matrix (RTM), Concept of Operations (CONOPS), Operational View diagrams, Interface Description Documents, Design Structure Matrix, Test Cases, Test Procedures, and deployment plans, among others, as outlined in Table 1. Furthermore, Interface Control Documents (ICD) may comprise multiple documents categorized by the type of interface—mechanical, software, and user interfaces—to resolve conflicts among team members from various disciplines involved in the same project. Given the multitude of components and subcomponents, developing an ICD for each can result in the creation and maintenance of thousands of documents throughout the system's lifecycle.

Systems engineers rely on these documents during various stages of the system development process, as they help facilitate the communication of critical information between teams. However, since the data presented in these documents lacks explicit dependencies, any change made in one document must be manually updated in all other affected documents. This manual process is not only time-consuming but also susceptible to errors. Furthermore, ensuring completeness and consistency and identifying conflicting or contradictory information becomes challenging when documents serve as the primary means of communication.

The ATO is the official management decision issued by a Designated Accrediting Authority (DAA) or Principal Accrediting Authority (PAA) to authorize the operation of an IS and to explicitly accept the residual risk to USG agency operations (including mission, functions, image, or reputation), assets, or individuals [3]. This risk can include loss of Personally Identifiable Information, Business Intelligence, general sensitive information/data, and Intellectual Property. The ATO is designed to determine if operationalizing a system outweighs the inherent risk it

introduces to the system owner. The ATO process is a standardized method of checks and balances to secure systems.

#### 1.4.1.1 THE CURRENT PRACTICE OF THE ATO PROCESS [4]:

In practice, the ATO has the following characteristics according to researchers from SEI.

- Authorizing applications is done just before operating the systems, and under ideal circumstances, the average time to get approval is ~6 months.
- The Program Manager (PM) is graded against the system's Key Performance Parameters (KPP) and their compliance with all regulations, along with cost and schedule parameters.
- The PM trades between cost, schedule, quality, and functionality. With each trade, residual risks occur.
- Someone must accept *all* residual risks associated with the system before placing it into Operations.
- The Authorizing Official (AO) is responsible for accepting information security risks through the RMF process.
- An ATO is usually good for 3 years but assumes no major changes to the system's cybersecurity posture will be made during that time.
- When changes occur, the AO may require a reassessment and reauthorization, which impacts the PM's cost and schedule and is contrary to Agile development methodologies [4].

The ATO for a Government system processing sensitive (including classified) information represents a decision based on data produced by an authorization process that involves a range of specialized practices. Achieving an acceptably low level of cybersecurity risk fundamentally involves identifying and assessing risks, selecting and prioritizing countermeasures (security

controls), implementing the security design, and testing, both initial and ongoing, to verify that threats are adequately mitigated [5]. The decision to grant a particular information system an ATO for a certain period is informed by an analysis of a system's security features and by test data from penetration attempts to determine if the risk of compromise is acceptable [6]. Current Government policy requires a continuous risk assessment during system operations to maintain the ATO [7]. Continuous risk assessment is known as Continuous Monitoring, and it can be accomplished in several ways. One of the most widely used for today's systems are vulnerability security scans using third-party applications and data log mining for abnormal system behavior. This data is analyzed by the security personnel and is required in intervals to maintain system accreditation, which is also determined by the security staff. Continuous Monitoring is illustrated later in Figure 11 - ATO Process Steps Block Definition Diagram.

Ultimately, an approving authority uses this risk information to decide whether the system/network in question will be allowed to operate for the next three years. There are inherent problems with C&A as the DoD has been performing it for decades; problems persist even when using the NIST RMF (described in 1.4.2). Many outside the information systems security field do not realize that C&A is just an *assurance* process that does not provide any security in and of itself. It provides only a level of confidence (i.e., assurance) that the system/network in question is compliant with the security requirements levied against it and attempts to quantify the risks associated with any security weaknesses identified by the testing [2].

Much of the current ATO process remains manual, particularly the substantial reporting and documentation associated with selecting, implementing, and testing controls. Many programs continue to use the Defense Information Assurance Certification and Accreditation Process (DIACAP) scorecard and allow up to 3 years before re-accreditation is required. DIACAP has

since been deprecated and replaced by the Risk Management Framework (RMF). “In 2014, the DoD started transitioning from the DoD Information Assurance Certification and Accreditation Process (DIACAP) to the RMF for the DoD IT. NIST Special Publication 800-37, "Guide for Applying the Risk Management Framework to Federal Information Systems," transforms the traditional Certification and Accreditation (C&A) process into the six-step RMF. The RMF provides a disciplined and structured process integrating information security and risk management activities into the system development lifecycle [8]. Due to both the scope and complexity of the process and the volume of systems requesting approval authorization, this can take several years. The cost is highly variable; our conversations with developers and accreditation experts indicate costs that vary by almost an order of magnitude and regularly exceed $1 million, depending on the AO assigned to accredit the system. During this time, vulnerability scanning and documentation must be continually updated to remain current upon AO review.” [9]. Even commercial vendors with commercially best-in-class security measures must undergo ATO to become accredited for use with DOD systems.

#### 1.4.1.2 STRUCTURE AND PROCESS OF THE ATO

The goal of this research is not to improve or optimize a trusted system's design. Instead, this research seeks to exploit the power of MBSE to reduce the time, especially in preparing documentation associated with the ATO, by automating much of the process, identifying and addressing inconsistencies, and reducing wasteful duplication of efforts. The ATO involves a series of steps in a rather extensive process designed to manage risk. It assures that the benefits of employing a system outweigh the cybersecurity risks. The ATO process consists of sequential steps: Prepare, Categorize, Select, Implement, Assess, Authorize, and Monitor, described in detail in the RMF in section 1.4.2 of this dissertation. Each step in the process takes time to create its

desired output. Each step has the potential to incur delays in fielding critical systems. By modeling each step and drilling into its specific elements, this dissertation seeks to reveal where time is allocated, and potential efficiencies exist to identify particular areas that should be prioritized for optimizing (lessening) the required time. The use of MBSE promises to reduce the time required for the ATO process significantly.

Securing vital US Government Systems is extremely important to ensure the safety and security of US assets, interests, and citizens. A formal process to guide the supporting sub-processes has been developed to guarantee better resilience and defense of Information Technology (IT) systems. This ATO process is a standardized method of checks and balances to secure systems through the various phases of the development process [10]. Its primary objective is to traceably insert security controls and methodologies into IT assets to reduce the risk of loss [10]. The standardized ATO process is not a "one size fits all: model because no two IT systems are identical. Each IT system is developed and deployed to meet a specific mission need, and the ATO process is adaptable to authorize the operation of the diversity of government IT systems. An ATO is an official declaration from a U.S. government agency authorizing using an application, platform, or product to operate within their secure network. Three unique ATOs can be granted, each providing operational availability based on security and temporal constraints. These determinations are primarily based on the system's maturity, its function, and the type of data it processes and handles. These three distinct ATOs illustrated in the diagram are described below [11].

- *Interim ATO*—A conditional ATO, generally in effect for six months, often during the development or prototype phase.

- *Initial ATO*—Must be done before the system "going live" and occur at least every three years after that.
- *Reauthorization*—This is due after three years or a notable change in the system's security risk level for a system already in production or operational use. Reauthorization can also be revisited at the discretion of the ATO Review Board.

#### 1.4.1.3 FUTURE NEEDS OF THE ATO

New needs have developed in the domain of government IT systems due to emerging technology and processes and the changing warfighting environment [12]. For example, the migration from monolithic, stove-piped systems (frequently filling a particular niche) to a modern cloud environment using technologies such as microservices, containers and container management, elastic scaling, new data storage techniques, and a host of solutions and services provided by industry has created the need to rethink the accreditation process [13]. Just as this mandate has driven modernization and made data and development more collaborative and available, a similar approach to system accreditation needs to use new processes and methodologies to compete in the ever-evolving landscape.

As Agile methodologies have become more common, frameworks such as Development, Security, and Operations (DevSecOps) [14] offer an opportunity to bring security into all development process phases. Instead of waiting until the end of the development cycle, which was common ATO practice, there is an opportunity to explore how DevSecOps, coupled with MBSE, could introduce security from the onset and continue through all development phases. In some parts of the Government space, this is already happening; security is factored in the early stages of the system development. However, the ATO continues to follow a document-intensive process, which slows the rapid development paradigm so desired by system users.

Similarly, the modern warfighting environment puts increasing emphasis on IT system security [15]. Ubiquitous internet connectivity and ongoing attacks in the cyber-domain has made the United States vulnerable to cyber warfare from strategic competitors like China, Russia, North Korea, and Iran. The result of these trends is a realization that the existing ATO process must be modernized to meet the changing landscape [16]. This is achieved by not abandoning the existing process but instead enabling it to be more efficient and less costly by enabling modern Systems Engineering methodologies, including MBSE.

There are efforts to modernize the ATO by adopting several strategies such as NIST 800-137 "Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations." Continuous monitoring in and of itself, does not provide a comprehensive, enterprise-wide risk management approach. Rather, it is a key component in the risk management process. Continuous monitoring activities contribute to helping AOs make better risk-based decisions, but do not replace the security authorization process [17].

ATO is an evolving process, but its importance to the current and future function of accreditation for USG IT systems is very high. All commercial, military IT systems that handle DOD data will have to undergo the ATO for the foreseeable future. Because of its importance, innovation applied to the ATO has the potential to realize substantial benefits for the security, integrity, and performance of accreditation of IT systems.

### 1.4.2 RISK MANAGEMENT FRAMEWORK

The ATO and accreditation process exists within the context of the National Institute of Standards and Technology (NIST) Risk Management Framework (RMF). RMF describes the guidance all federal agencies must follow to secure, authorize, and manage information systems

and specifies a process for initially securing and then integrating constant monitoring [18]. In 2014, the DoD started transitioning from the DoD Information Assurance Certification and Accreditation Process (DIACAP) to the Risk Management Framework for the DoD IT (RMF). NIST Special Publication 800-37, "Guide for Applying the Risk Management Framework to Federal Information Systems", transforms the traditional Certification and Accreditation (C&A) process into the six-step RMF. The RMF provides a disciplined and structured process integrating information security and risk management activities into the system development lifecycle [8]. Being granted an ATO demonstrates that a federal agency has undergone this federally approved process to protect an IT system from cyberattacks, security breaches, malware, and phishing attempts [19]. Choosing the right security controls is contingent upon the specific needs, objectives, and acceptable level of risk for the program, as all systems inherently carry some level of risk. Additionally, it's important to consider the availability of resources to reach an acceptable level of security before operations commence. According to the DoD Systems Engineering Guidebook, February 2022, there are five risk management process activities that need to be conducted:

- Risk Planning – "What is the risk management process?"
- Risk Identification: What can go wrong? Are there emerging risks based on Technical Performance Measure trends or updates?
- Risk Analysis – What is the likelihood of the undesirable event occurring and the severity of the consequences?
- Risk Mitigation – Should the risk be accepted, avoided, transferred, or controlled?
- Risk Monitoring – How has the risk changed? [20]

The RMF is a process designed to enable the ATO, which also consists of sequential steps: *Prepare*, *Categorize*, *Select*, *Implement*, *Assess*, *Authorize*, and *Monitor,* which are as follows:

- Prepare – gather information about the system, including its primary functions as they map to requirements.
- Categorize – determine the criticality of the system based on potential adverse impacts of a compromise on the using activity.
- Select – choose the baseline security controls of the system.
- Implement – incorporate the security controls in the system design.
- Assess – evaluate the security controls to determine their effectiveness.
- Authorize – issue approval for the system to operate (approved ATO).
- Monitor – re-evaluate the system over time to ensure an acceptable risk posture is maintained.

Figure 1 illustrates the relationships and interactions among these steps in the form of a SysML Activity Diagram.

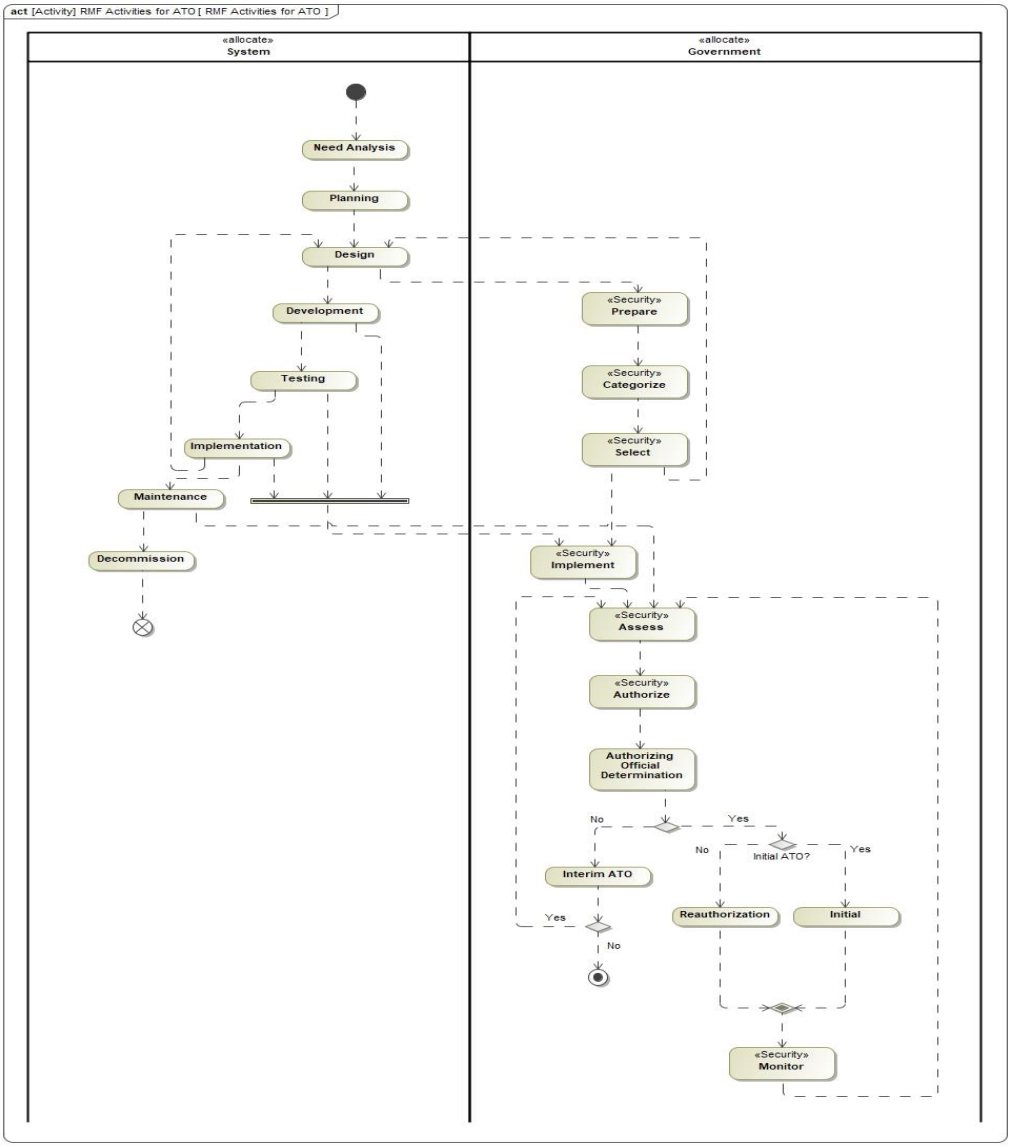



*Figure 1– Activity Diagram representing the RMF activities required for ATO*

To further the analysis of the ATO, a Sequence Diagram is presented as Figure 2 which provides a timing analysis based on ideal situations. The Sequence Diagram shows that under these conditions, the ATO is twelve months without documentation development and review.

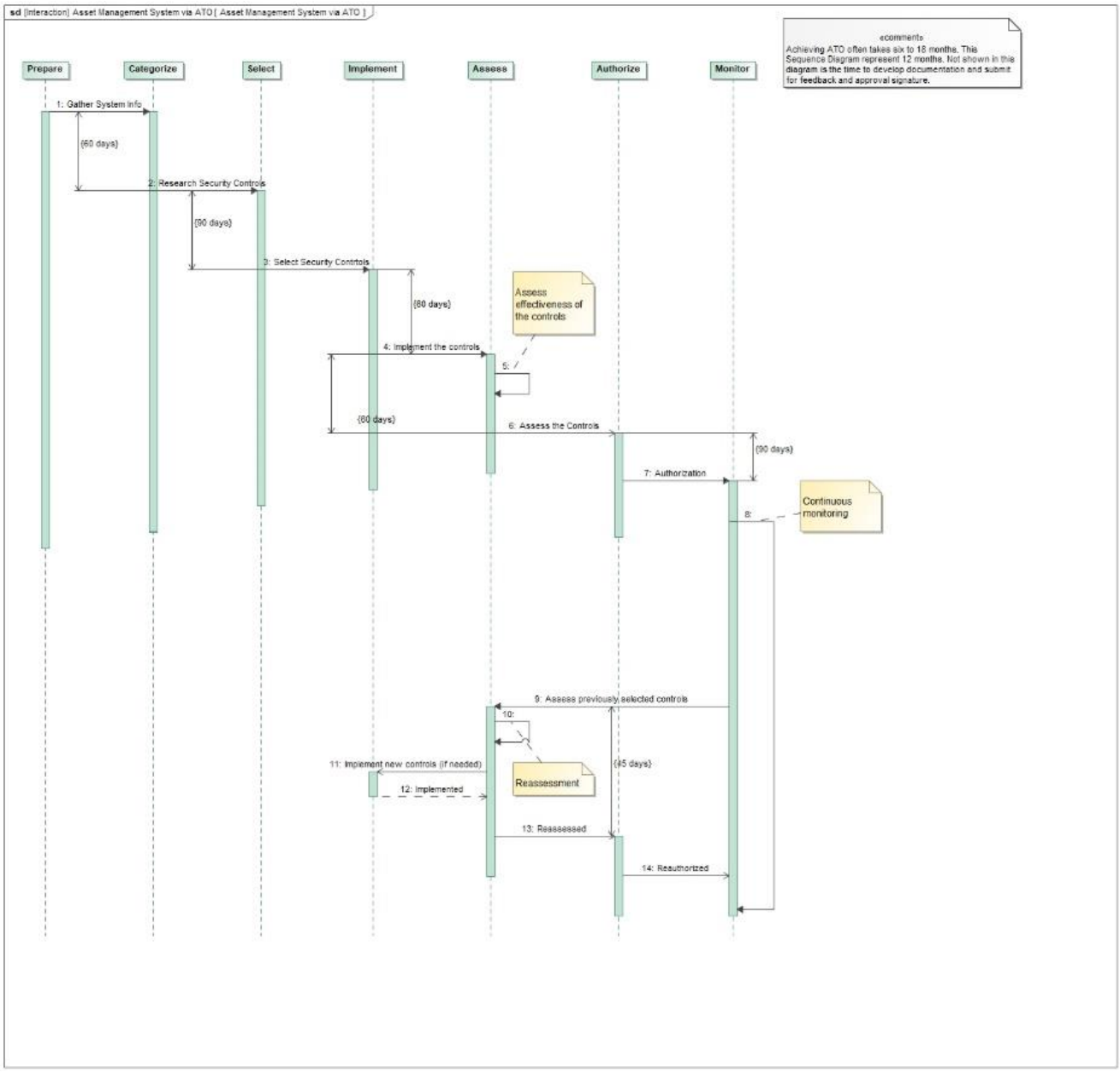



*Figure 2 - Sequence Diagram with ATO Timing Analysis*

Although the ATO and RMF are ubiquitous in government IT systems development, a set of problems have been identified with the RMF and ATO as they are currently being executed. To reiterate, some of the challenges are:

- As stated previously, every IS has inherent risks associated with it, but the RMF does not acknowledge the presence of residual inherent risk.

- The Program Manager (PM) is graded against the system's KPP, their compliance with all regulations, and cost and schedule parameters. However, the PM is often not graded against success in cyber defense because of the difficulty of its assessment.
- The PM makes many ongoing trades between cost, schedule, quality, and functionality. With each trade, residual risk occurs, which is often not acknowledged as a tradeoff.
- The RMF does not allocate risk to organizations, and someone or some organization must accept the residual risk associated with the system before placing it into Operations.
- The Authorizing Official (AO) is responsible for accepting information security risks, which is done through the RMF process.
- An ATO typically authorizes an IS for three years but assumes no significant changes to the system's cybersecurity posture will be made during that time. This may not be an accurate assumption in fast-moving operational domains (Cribbs, 2002).
- When system changes do occur, the AO may require a reassessment and reauthorization, which often impacts the PM's cost, schedule, and ability to deliver capabilities [8].

It is not clear that a model-based ATO process will be able to solve all problems that are associated with ATO and RMF. This research asserts that some of the cost and schedule benefits that are asserted to accrue, an MBSE-enabled Systems Engineering processes has the potential to reduce the costs of the tradeoffs in the larger RMF context.

Being granted an ATO demonstrates that a federal agency has undergone a federally approved, detailed process to protect an IT system from cyberattacks, security breaches, malware, and phishing attempts. Many federal IT systems are required to obtain an ATO to process government data, and federal regulations recommend that agencies follow the Risk Management Framework (RMF) to become authorized [19].

Table 1 is the result of an analysis of the deliverables often required in a systems development process. Some of the documentation is system-specific, covering its need, design, and general system characteristics. Some of the other documentation is security and test-related to demonstrate and report how the system handles and mitigates threats. The documentation was then categorized based on which step of the RMF they are most likely to be associated. In total, a typical Government IS will require somewhere on the order of 41 documents; nine of which are required for accreditation, and five that are often deemed necessary.

*Table 1 – ATO Steps and Expected Documentation for System Delivery*

| ATO Step | Non-ATO Documentation – Additional but Recommended – Often Requested | ATO Documentation Package - Required | Common in ATO Package - Additional | Total Amt of Docs |
|---|---|---|---|---|
| Prepare | Concept of Operations<br>Software Development Plan<br>Software Installation Plan<br>Software Transition Plan<br>Operational Concept Description<br>Software Test Plan<br>System Requirements | | Privacy Impact Assessment<br><br>Privacy Threshold Assessment<br><br>Incident Response Plan<br><br>Disaster Recovery Plan<br><br>ATO Boundary Diagram | **41** |
| Categorize | System / Subsystem Design Description | System Definition Document | | |
| Select | System / Subsystem Specification<br>Software Requirements Specification<br>Interface Requirements Specification<br>Software Product Specification | System Security Plan | | |
| Implement | Software Design Description<br>Interface Design Description<br>Database Design Description<br>Software Test Description<br>Software Test Procedure | Updated System Security Plan<br>Status Report | | |
| Assess | Software Test Report<br>Software Version Description<br>Requirements Traceability Matrix | Security Assessment Report<br>Security Assessment Plan | | |
| Authorize | Software User Manual<br>Software Center Operator Manual<br>Software Input/Output Manual<br>Computer Operation Manual<br>Computer Programming Manual<br>Firmware Support Manual | POA&M<br>Risk Assessment | | |
| Monitor | Continuous Monitoring Plan | Monitor Strategy Document | | |

The primary goal of the RMF is to provide three objectives: Confidentiality, Integrity, and Availability (CIA). These objectives are often referred to as the "CIA Triad" [12].

- Confidentiality - "preserving authorized restrictions on access and disclosure, including means for protecting personal privacy and proprietary information."

- Integrity - “guarding against improper information modification or destruction and includes ensuring information nonrepudiation and authenticity.”
- Availability - “ensuring timely and reliable access to and use of information.” [12]

Then based on those the following questions should be considered.

- What is the worst possible outcome if all of the *confidentiality* of the system is lost? i.e.
    - What if all of the data in the system is exposed to the public?
- What is the worst possible outcome if all of the *integrity* of the system is lost? i.e.,
    - What if an error makes it into the data?
    - What if an update to the data is lost?
- What is the worst possible outcome if all of the *system's availability* is lost? i.e.
    - What if the system has downtime? (Open Control, 2022)

In summary, the ATO process is a well-developed and commonly executed process to manage the risk of loss for government IT systems in the context of an existing RMF. The challenges associated with contemporary IT system development, the digital transformation of IT system architecture and design, and the heightened risk of cyber incidents frame the context in which modern MBSE methods and tools can enhance the efficiency and cost-effectiveness of the ATO process.

### 1.4.3 SYSTEMS ENGINEERING

The International Consortium of Systems Engineers (INCOSE) defines Systems Engineering as “a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods [21].” The Systems Engineering Guidebook published by the DoD, February 2022, defines Systems Engineering in greater detail.

*"... the primary means for determining whether and how the challenge posed by a program's requirements can be met with available resources. It is a disciplined learning process that translates capability requirements into specific design features and thus identifies key risks to be resolved. Our prior best practices work has indicated that if programs apply detailed SE before the start of product development, the program can resolve these risks through trade-offs and additional investments, ensuring that risks have been sufficiently retired or that they are clearly understood and adequately resourced if they are being carried forward."*

*"...SE planning, as documented in the Systems Engineering Plan (SEP), identifies the most efficient path to deliver a capability, from identifying user needs and concepts through delivery and sustainment."*

The term Systems Engineering dates to Bell Telephone Laboratories in the early 1940s [Schlager, 1956; Hall, 1962; Fagen, 1978]. Fagen [1978] traces the concepts of systems engineering within Bell Labs back to the early 1900s and describes major applications of systems engineering during World War II. Hall [1962] asserts that the first attempt to teach systems engineering as we know it today came in 1950 at the Massachusetts Institute of Technology (MIT) by Mr. Gilman, Director of Systems Engineering at Bell [21].

Hall [1962] defined Systems Engineering as a function with five phases:

- System studies or program planning.
- Exploratory planning, which includes problem definition, selecting objectives, systems synthesis, systems analysis, selecting the best system, and communicating the results.
- Development planning, which repeats phase 2 in more detail.
- Studies during development, which includes the development of parts of the system and the integration and testing of these parts; and
- Current engineering, which is what takes place while the system is operational and being refined.

The RAND Corporation was founded in 1948 by the United States Air Force and created systems analysis, which is an important function of systems engineering [21].

The Department of Defense entered the world of Systems Engineering in the late 1940s with the initial development of missiles and missile-defense systems [Goode and Machol, 1957]. Currently, Systems Engineering (SE) plays a vital role in producing assets such as ships, aircraft, software, vehicles, and other complex components of the DoD. All these systems require accreditation before becoming operational. That process is the ATO. As a result of heuristic understanding of the discipline, it has in the past been nearly impossible to quantify the value of SE to programs (Sheard 2000). Yet both practitioners and managers intuitively understand that there is indeed inherent value and will typically incorporate some SE practices in every complex program. The principle impact of the systems engineering paradigm is to reduce risk early [22].

According to the Sandia Report SAND2016-2607 – "Systematic Literature Review: How is Model-Based Systems Engineering Justified," no case studies were found that compared an MBSE approach side-by-side with a Document Based Systems Engineering (DBSE) approach in a controlled experiment. The closest to a side-by-side comparison found was a single case study referred to by Honour in his thesis, "Systems Engineering Return on Investment," conducted by The Boeing Company in 1995 – well before MBSE. The case study referenced by Honour was authored by [23] and gave an example of how the Boeing Company justified an SE approach by comparing improvements gained from employing three various levels of systems engineering processes on three similar projects conducted simultaneously. Honour summarizes the case study below and illustrates the performance between the three projects in Figure 3: A unique opportunity occurred at Boeing in which three roughly similar systems were built at the same time using different levels of systems engineering. The three systems were Universal Holding Fixtures (UHF) used for manipulating large assemblies during the manufacture of airplanes. Each UHF was of a size on the order of 10' x 40', with accuracy on the order of thousands of an inch. The three varied

in their complexity, with differences in the numbers and types of sensors and interfaces. Three similar projects were run in parallel. Each had varying degrees of SE disciplines implemented – from nearly none to high. The two projects using SE were delivered more than twice as fast. The project using the highest level of SE was delivered nearly three times faster and had the highest quality [24].


*Figure 3 - Three Systems Engineering Efforts Compared [22]*

#### 1.4.3.1 DIGITAL ENGINEERING

DE is an integrated digital approach that uses authoritative sources of systems data and models as a continuum across disciplines to support lifecycle activities from concept through disposal [25]. "Digital engineering describes a holistic approach to the design of a complex system: Design using models/data instead of documents, integration of data across models, and the culture change across project teams to realize significant risk reduction on construction cost and schedule [26]." In an applied application, DE provides the ability to create "living" virtual models of complex IT systems. Thus, it provides more effective ways to test potential solutions to modernize IT infrastructure. It helps gain critical insights on how changes in IT systems might impact interactions with a broader circle of weapons, satellites, communications and other operating systems [27]. Digital Engineering uses and integrates digital models and the underlying data to support the development, test and evaluation, and sustainment of a system [28].

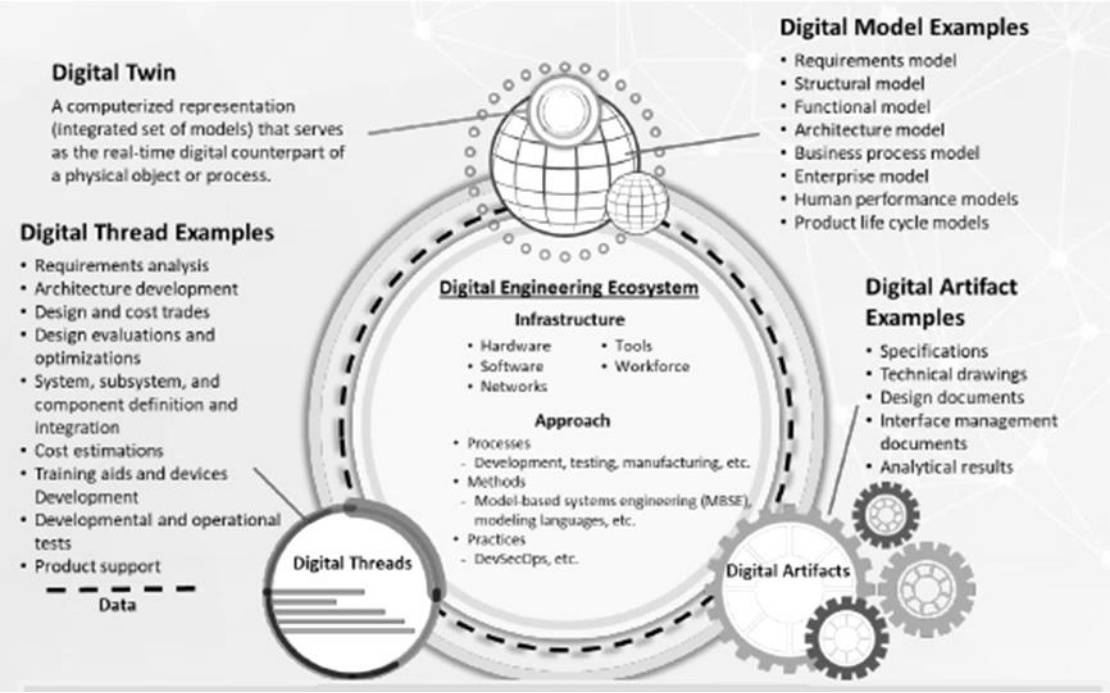



*Figure 4 - Overview of Digital Engineering [29]*

As depicted above in Figure 4 - Overview of Digital Engineering , MBSE is a discipline of DE. In traditional document-centric engineering, modeling relied on technical drawings, notations, and calculations. MBSE applies DE to traditional engineering processes to modernize and streamline systems engineering in an optimized digital format. DE provides the concepts of using digital data and the underlying computer processing required to simulate systems, and MBSE specifically applies these principles to object and system models [30].

Another important aspect of DE is the use of digital twins and digital threads, which are used to represent assets of the system or components digitally. “A digital twin is a digital replica of a physical object or system, complete with all the design and operational data of the physical object, including geometry, performance data, and behavior models. The purpose of a digital twin is to simulate the behavior of equipment in real time, allowing engineers and operators to monitor performance and identify system issues/anomalies. A digital thread is a digital representation of a product’s lifecycle, from design to manufacturing to maintenance and beyond, providing a

seamless flow of data that connects all aspects of the lifecycle. The purpose of a digital thread is to provide a complete and transparent view of manufacturing systems, enabling efficient collaboration and decision-making across all stages of the process" [31].

#### 1.4.3.2 MODEL-BASED SYSTEMS ENGINEERING

MBSE is a systems engineering paradigm that focuses on the use of models to perform systems engineering tasks that are traditionally done using documents. Implementing MBSE requires using a modeling language, modeling methods, and modeling tools [32]. MBSE is a category of DE where models are created to represent real-world systems through design, development, analysis, simulation, and testing (the entire life cycle). MBSE modeling languages, methods, and tools offer the possibility to implement rigorous modeling techniques that incorporate traditional systems engineers' best practices to develop a central, unambiguous, organized, and precise model of the system [33]. MBSE is an approach to system engineering that relies on graphical models to conceptualize, design, analyze, and document complex systems. It involves using:

- modeling languages like SysML or Unified Modeling Language (UML),
- modeling tools,
- analysis techniques,
- and integration methods.

The main goal of MBSE is to improve communication, understanding, and decision-making throughout the system development lifecycle [34].

The three pillars of MBSE are models, methods and tools:

- **Models**: The central component of MBSE, models represent different aspects of the system being developed, including its structure, behavior, requirements, interfaces, and

interactions. Models serve as a means of visualizing, analyzing, and documenting the system throughout its lifecycle.

- **Methods**: MBSE employs various modeling and analysis techniques, methodologies, and processes to develop, analyze, and validate system models. These methods help engineers and stakeholders understand system requirements, define system architecture, simulate system behavior, and make informed decisions.
- **Tools**: MBSE relies on specialized software tools that support the creation, manipulation, analysis, and documentation of system models. These tools provide features such as graphical modeling editors, simulation capabilities, version control, and integration with other engineering tools, enabling efficient collaboration and communication among stakeholders [34].

The key benefits of MBSE include increased clarity and collaboration in the design process, increased accuracy of specifications, improved product quality, enhanced knowledge capture, improved traceability, faster system development, and reduced overall costs [35]. Using models helps clarify the system design, improves communication, and facilitates a better understanding of the system requirements [36]. Similarly, Dr. John M "Mike" Borky and Dr. Thomas H. Bradley of Colorado State University find that MBSE helps to:

- Ensure rigor, repeatability, and producibility in SE processes
- Promote quality, completeness, and correctness in system designs
- Reduce risk in requirements analysis, design, integration, test, and other activities
- Enhance communication and synchronization of activities across organizations and disciplines. [32]

Figure 5 and Figure 6 illustrates at a high level of abstraction, the relationship between systems engineering activities and modeling under an MBSE paradigm, as conceptualized by different authors.

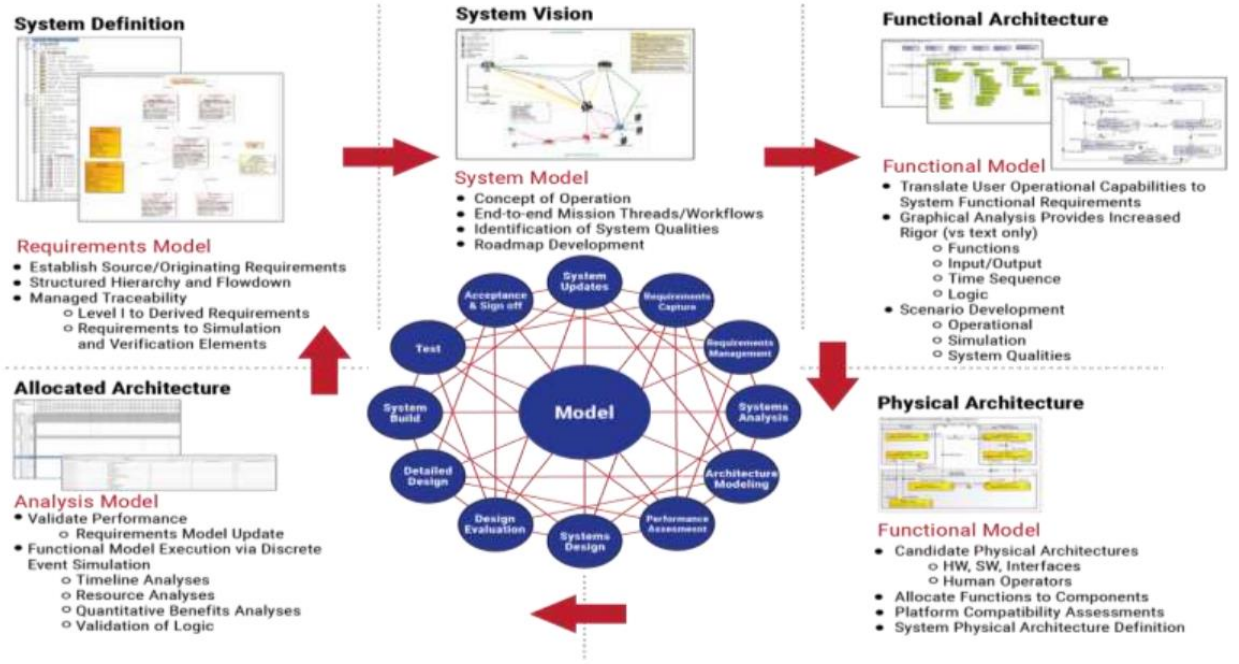



*Figure 5 - High-Level Model of MBSE [37]*

Modeling within a MBSE context does not aim to create a perfect replica of the actual IT system, which is typically associated with developing a Digital Twin (a topic discussed later in this section). Instead, MBSE models are designed to provide knowledge and feedback more quickly and cost-effectively than implementation alone. They enable the simulation of complex system and system-of-system interactions with the appropriate level of detail, helping to accelerate learning. In practice, engineers use models to gain knowledge and to serve as a guide to system implementation. In some cases, engineers use them to directly build the actual implementation. This is done after several iterations of the model [38].

Figure 6 below shows where MBSE aligns with the traditional Systems Engineering activities. The MBSE development activities and deliverables are highlighted in yellow on the right.

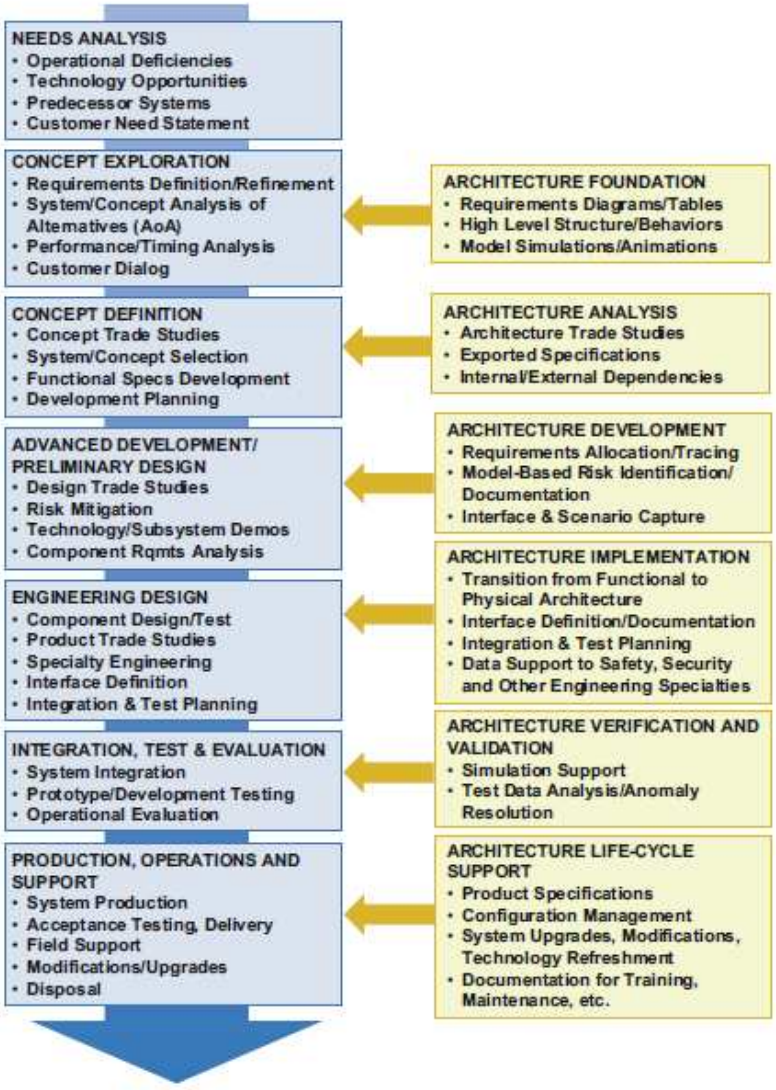



*Figure 6 - MBSE Support to Systems Engineering Activities [32]*

The need to shift to Model-Based Development (MBD) primarily stems from the inherent complexity that the manual coding process brings about during development (Els, 2019). This is specifically true for our example system. Not only is the size of the code a concern, but the complex application of the code also makes it quite difficult to maintain the functional structure of the

overall system software. In manual coding, the software developer is often more focused on the code instead of its function. Furthermore, large code bases are also difficult to port to other Independent Development Environments (IDE) and controllers if the need for hardware platform migration arises. Hence, the biggest challenge faced by software engineers while working on these complex systems is to shorten development cycles and reduce development and testing time while ensuring system integrity. When used with simulation tools, MBD models enable rapid prototyping, software testing, and verification. Not only is the testing and verification process enhanced, but also, in some cases, hardware-in-the-loop simulation can be used with the new design model to perform testing of dynamic effects on the system more quickly and much more efficiently than with traditional software design methodology [39]. Applying Model-Based Development can realize average cost savings of between 25 to 30 percent and time savings of 35 to 40 percent [39]. Similarly, applying MBSE provides opportunities to capture and harmonize information from multiple disciplines within an integrated digital model of the system. Since dependencies across multiple disciplines are explicitly addressed in the model, a change made in one area is automatically carried across to all the related software subsystems. This feature makes it possible to maintain consistent and up-to-date information in the integrated digital model. In addition, completeness, consistency, traceability, and contradiction checks can be performed, while the ability of MBSE to identify defects early in the system lifecycle also results in significant cost savings" [39].

The research, simulation, and experimentation defined in this dissertation outlines the benefits of MBSE on the ATO process. Those areas are discussed in more detail through research questions and tasks. However, there are items outside of the scope of MBSE that would be considered for future research or investigation. These include:

- Lack of clear communication between the organization and the authorizing agency,
- Incorrect or incomplete documentation provided to the authorizing agency,
- Noncompliance with federal regulations and security policies,
- Weaknesses in information security controls or processes,
- Insufficient evidence of proper risk management and testing,
- Incomplete or inaccurate reporting of incidents or vulnerabilities,
- Inconsistent or unclear criteria for approving or denying an ATO request,
- Lack of resources or funding for the authorization process,
- Inadequate training or awareness of security policies and practices among personnel,
- Failure to comply with security policies and requirements after obtaining an ATO.

One approach that has gained a lot of traction is the "digital twin." The Idaho National Laboratory (INL) defines a digital twin as a virtual model that mirrors a physical asset and is used to predict future behavior. Digital twins are made up of a combination of connected data, sensors, instrumentation, artificial intelligence, and online monitoring. They use real-time communication to track and trend both simulated and measured asset information.

Implementing an MBSE strategy comes with its own set of challenges. For some participants in the acquisition process, using models to manage the technical baseline can be less intuitive compared to traditional documents. This can lead to models being treated merely as descriptive end products rather than as the foundation of the technical management process, which undermines the intent of MBSE [35]. Additionally, the landscape is complicated by the presence of competing tools and languages; however, Cameo has seemingly emerged as the preferred modeling tool alongside the use of SysML. Furthermore, adopting MBSE requires either extensive training or the immediate hiring of highly skilled engineers, along with the acquisition of supportive software

tools. The initial investment can be substantial and should be carefully considered when evaluating a transition. While there are numerous examples of successful implementations, there is growing concern about how MBSE interacts with non-technical disciplines. The figure below illustrates the rate of innovation adoption, which is relevant to the transition to MBSE from a traditional DB-ATO environment.

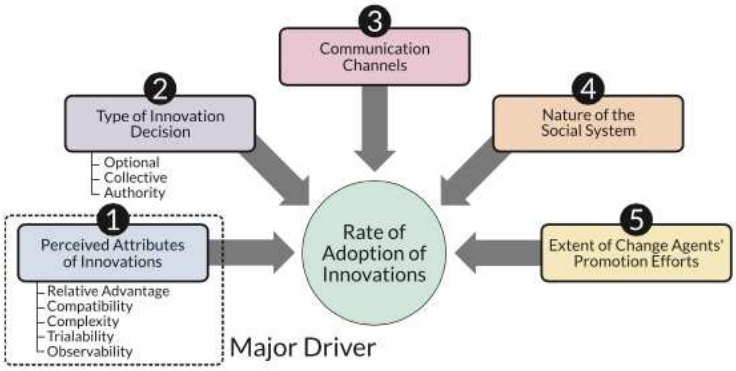



*Figure 7 - Variables that Affect the Adoption Rate of Innovative Technology [40]*

#### 1.4.3.3 SYSTEMS MODELING LANGUAGE (SYSML)

SysML is a general-purpose system architecture modeling language for Systems Engineering applications [41]. SysML supports the specification, analysis, design, verification, and validation of a broad range of systems. These systems may include hardware, software, information, processes, personnel, and facilities. SysML is a variant of UML 2 and functions as a UML 2 Profile. A UML Profile customizes the language through three mechanisms: Stereotypes, Tagged Values, and Constraints. It is the enabling technology for modern MBSE and is used to support the modeling for this dissertation. SysML has practical semantics for capturing process steps, data object creation and exchange, organizational roles and responsibilities, personnel skills and qualifications, decision and synchronization points, timing, constraints, and other process features. The Object Management Group (OMG) defines SysML as:

*"A general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. In particular, the language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models."*

SysML can represent the following aspects of systems, components, and other entities:

- Structural composition, interconnection, and classification;
- Flow-based, message-based, and state-based behavior;
- Constraints on the physical and performance properties;
- Allocations between behavior, structure, and constraints; and
- Requirements and their relationship to other requirements, design elements, and test cases [42].

SysML supports the practice of MBSE that is used to develop system solutions in response to complex and often technologically challenging problems [42]. SysML includes nine diagrams, as shown in the taxonomy in Figure 8. Each diagram kind is summarized below, along with its relationship to UML diagrams:

- *Package diagram:* This diagram presents the organization of a model in terms of packages that contain model elements (same as UML package diagram).
- *Requirement diagram:* This diagram presents text-based requirements and their relationships to other requirements, design elements, and test cases to support requirements traceability (not in UML).

- *Activity diagram:* presents flow-based behavior indicating the order in which actions execute based on the availability of their inputs, outputs, and control, and how the actions transform the inputs to outputs (modification of UML activity diagram).
- *Sequence diagram*: This diagram presents behavior in terms of a sequence of messages exchanged between systems or parts of systems (the same as a UML sequence diagram).
- *State machine diagram*: This diagram presents an entity's behavior in terms of its transitions between states triggered by events (it is the same as a UML state machine diagram).
- *Use case diagram*: This diagram presents functionality in terms of how a system is used by external entities (i.e., actors) to accomplish a set of goals (the same as a UML use case diagram).
- *Block definition diagram*: This diagram presents structural elements, called blocks, and their composition and classification (modification of UML class diagram).
- *Internal block diagram*: This diagram presents interconnections and interfaces between the parts of a block (modification of UML composite structure diagram).
- *Parametric diagram*: This diagram presents constraints on property values, such as F=m*a, used to support engineering analysis. It is primarily concerned with modeling a system's quantitative constraints and parameters (this diagram is not part of UML) [42].

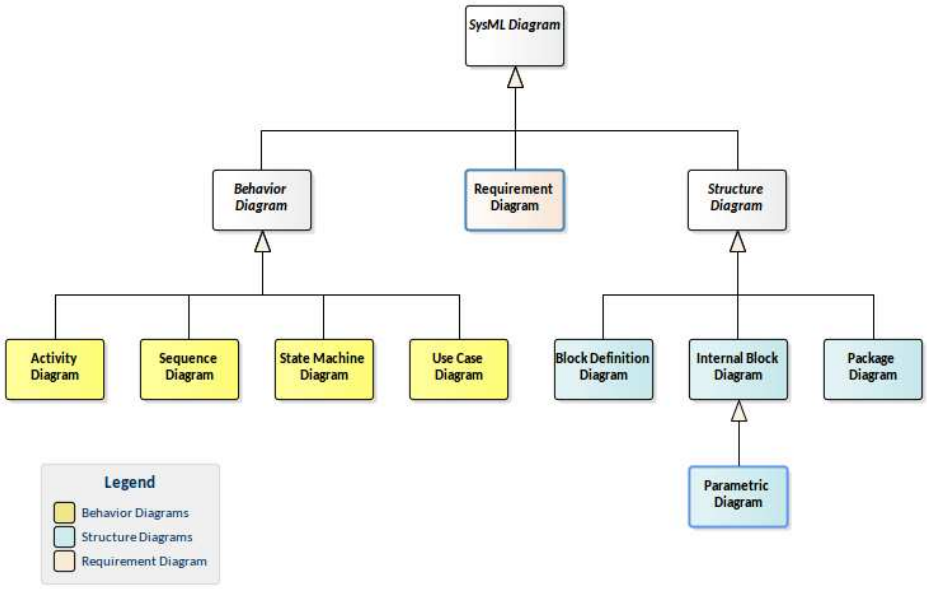



*Figure 8 - SysML Diagram Taxonomy (cameomagic.com)*

These diagram types are grouped into three main categories: Structural, Behavioral, and Requirement, each with its own unique set of elements. These elements are linked together to create a complete system model that captures the intricacies of a complex system. Structural diagrams capture the physical and logical structure of a system, including its components, interfaces, and relationships. Behavioral diagrams capture the dynamic behavior of a system, including how it responds to various stimuli and events. Finally, Requirement diagrams capture the requirements that a system must fulfill, whether they are functional or non-functional [43].

## 1.5 PROPOSED SOLUTION

Drawing on field experience, this dissertation aims to develop and evaluate a model-based approach for accrediting USG systems, specifically aligning with the ATO process. The objective is to modernize the ATO process by tackling three key areas of concern and executing a series of tasks to address these issues. The proposed system is assumed to adhere to the traditional lifecycle model outlined below.

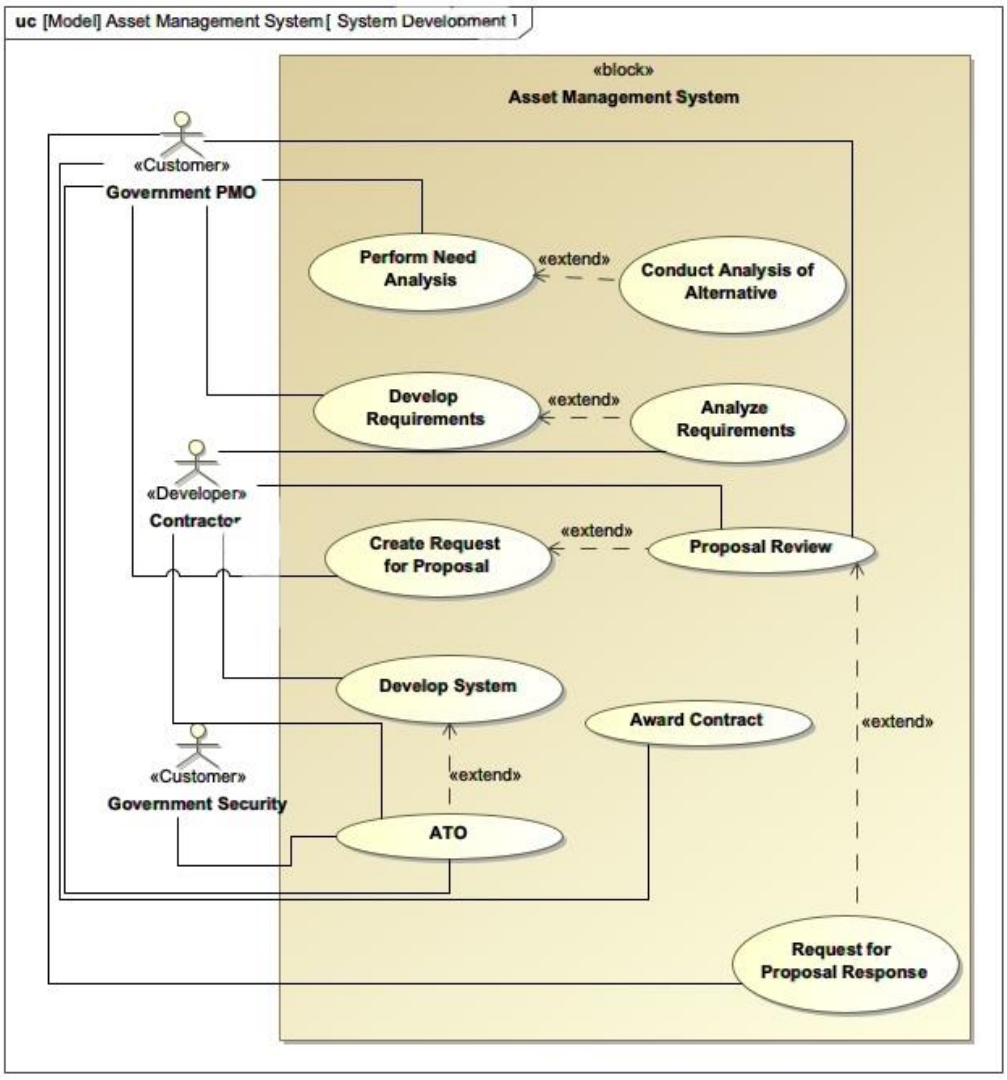



*Figure 9 - Traditional Government IS Development Model*

With a clear understanding of the current landscape and the potential of MBSE processes and tools to enhance the development and accreditation process and its products, there can now formulate a research agenda aimed at advancing the knowledge of how and why MBSE tools and methods can lead to improved performance and cost-effectiveness in the ATO process.

This dissertation offers a fresh perspective and metrics regarding the impact of MBSE on the ATO. It seeks to evaluate whether the integration of MBSE within the ATO framework can yield a return on investment and provide essential support to program stakeholders, deploy systems more quickly and with higher quality while reducing inconsistencies and cost.

The following sections present the research agenda for this dissertation, structured as a series of related research questions and tasks.

# CHAPTER 2 – RESEARCH AGENDA

This chapter outlines the research agenda, and the methodologies employed to address a series of research questions through specific tasks. The first step involved creating a model of the ATO process, as defined by a series of steps outlined in the RMF to achieve accreditation. Subsequently, a cloud-native Asset Management system model was developed using tools currently available from the U.S. Government (USG). These two models formed the foundation of the research, representing realistic system entities, processes, and timelines to evaluate the impact of MBSE on the ATO process. Additionally, interviews were conducted, and the research agenda underwent peer review by subject matter experts.

## 2.1 RESEARCH METHODS

The main objective of this research is to develop a model of the ATO and a cloud-native transactional system, referred to as the Asset Management System, which can serve as an exemplar of a real-world government asset. Once created, these models could be customized and utilized to illustrate the effects of MBSE on the ATO accreditation process for similar systems. This dissertation aims to support informed decision-making regarding the adoption of MBSE as the preferred framework for all system accreditation within the U.S. Government.

This chapter offers a comprehensive overview of the methodology employed in the subsequent case studies of this research. Following the explanation of the proposed solution in Section 1.5, the method comprises three distinct components. The proposed process will be validated through the execution and expansion of these components via case studies while incorporating expert feedback, real-world comparisons when feasible, and direct analytical assessments of the solution's quality. The structure of the dissertation consists of three main

chapters, each initially framed as questions regarding how the ATO can be enhanced by leveraging MBSE. The hypothesis posits that MBSE will improve efficiency by reducing documentation errors, reducing system costs through reuse, and reducing the time required to deploy mission-critical systems. These three hypotheses manifest in three phases of the system's development lifecycle, as illustrated in Figure 10. Three Research Questions (RQs) have been formulated to attain the research goal.

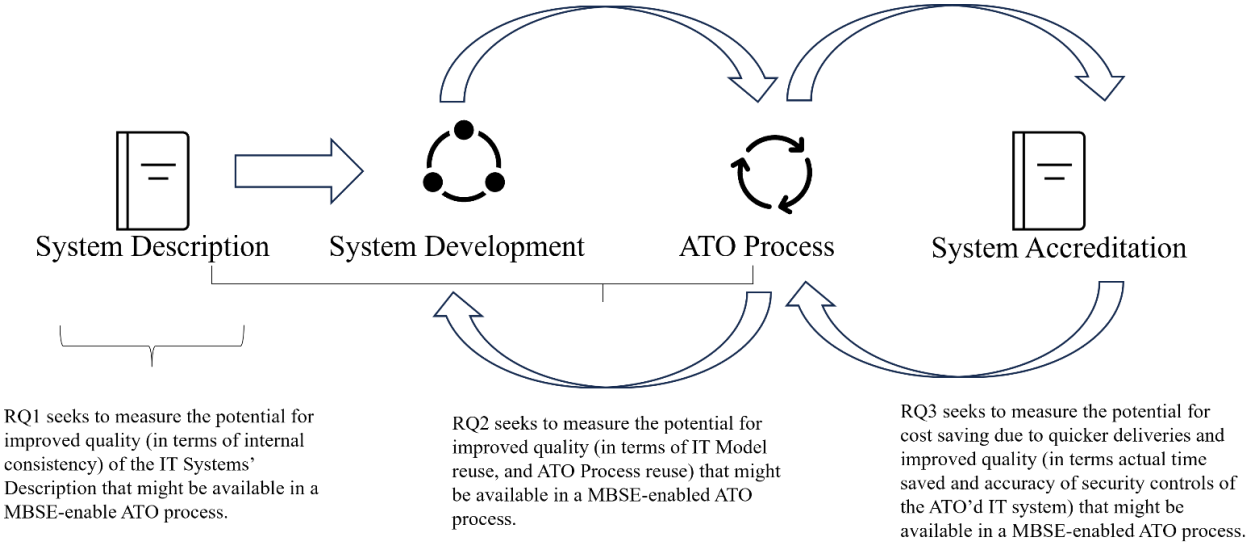



*Figure 10 - Scope of Research Questions*

## 2.2 DESCRIPTION OF RESEARCH QUESTION ONE (1): INCONSISTENCIES

Inconsistencies in documentation and version control can lead to confusion and increase the costs of a program. This RQ seeks to understand if an MBSE model and tool can effectively provide error correction in real or near real-time and can mitigate areas prone to confusion due to inconsistencies. This will promote efficient development and deployment. Table 2 illustrates the errors in ATO technical documentation. These types of errors are difficult to correct in a document-centric ATO process. Modern MBSE tools and models have means to detect and correct these

types of errors. RQ 1 seeks to measure and validate the effectiveness of these types of tools in the ATO document generation process in reducing inconsistencies.

*Table 2 - Examples of Problems in Documentation*

| Documentation Problems | Example |
|---|---|
| Bad references | Document A references Document B sub-paragraph X which is incorrect |
| Vagueness | The document lacks sufficient details to be effective |
| Bad mapping to requirements | Requirement A is traced to a functional test case but has the wrong reference (e.g. Req A.1.1, but it should be Req A.1.2) |

### 2.2.1 RESEARCH QUESTION ONE (1)

*Does an MBSE model in the development lifecycle and ATO process reduce typical inconsistencies in traditional document-centric systems engineering?*

### 2.2.2 TASKS FOR RESEARCH QUESTION ONE (1)

To answer Research Question One (1), the following associated tasks were developed to help the author reach an informed conclusion.

- Manually survey Requirements documentation for references and note inconsistencies. This can be especially true with requirements documentation referring to itself or other programmatic documentation.
- Develop requirements in Cameo and intentionally reference resources that do not exist in the model or are misspelled, etc., to demonstrate the error detection capabilities of MBSE.
- Provide examples through research of how a system without proper version control can lead to inconsistencies (email, file system, etc.).
- Create a model where centralized data reduces version control inconsistencies to demonstrate MBSE's use of configuration management.

## 2.3 DESCRIPTION OF RESEARCH QUESTION TWO (2): REUSE

Reuse is often encouraged in software development programs, specifically code reuse. This commonly keeps costs lower while promoting quality. This research asserts that the reuse of Information Systems and ATO process models and artifacts may improve the quality and consistency of the ATO. This question will provide evidence to understand if the same is true for MBSE models and if reuse is beneficial in keeping with cost projections for the program while the quality continues to improve.

### 2.3.1 RESEARCH QUESTION TWO (2)

*Does reusing an MBSE model across various similar programs create an improved quality process and an improved product in less time? In the context of the ATO process means a more streamlined and lean procedure with improved "ilities" (e.g., maintainability, portability, reliability, operability, et. al.) by providing internal consistency and integrity (no loose ends and minimized points and interfaces of failure). How does this affect documentation?*

### 2.3.2 TASKS FOR RESEARCH QUESTION TWO (2)

Building upon the methodological framework established in addressing Research Question One, this analysis extends and adapts those techniques to examine Research Question Two while maintaining consistency in analytical approach, by identifying and executing tasks to come to an informed conclusion. Those tasks are as follows.

- Compare the architecture to show continued maturation with requirements.
- Compare the time to create programmatic documentation traditionally vs. using a model tool such as Cameo to generate them automatically from templates.

- Research at what point it is the most cost-effective to invest in generating templates for document generation instead of manually. i.e., investing in template creation for a project of short duration and delivery is likely, not cost-effective.
- Research the general cost for required documentation for a program and ATO to show that reducing documentation via MBSE lowers system cost.

## 2.4 DESCRIPTION OF RESEARCH QUESTION THREE (3): SAVINGS

Empirical studies have demonstrated MBSE's capacity to significantly reduce both development costs and deployment timelines in software/system engineering projects. Therefore, does reuse of a system model increase the speed to delivering an operational solution? Most software system solutions in the federal space are now operating in the cloud using standard services and components. Will reuse of already accredited IS models that share a common cloud architecture promote faster deployments because of the existing model and already identified security controls? For example, if system A is a cloud-based software solution that was built, granted an ATO, and deployed operationally using a model, would system B benefit from reusing that model as a basis for its development and authorization? Therefore, it also reduces costs as a byproduct of using a previously accredited model.

### 2.4.1 RESEARCH QUESTION THREE (3)

*Can MBSE reduce the overall cost of an ATO program by enabling quicker releases? Does the reuse of models promote more timely releases, and does the accuracy and predictability of the security controls for similar threat scenarios based on artifacts lead to less overhead?*

### 2.4.2 TASKS FOR RESEARCH QUESTION THREE (3)

Lastly, continuing to follow the approach outlined for the previous two questions, Research Question Three (3) also has associated tasks that were developed to come to an informed conclusion. Those tasks are as follows.

- RT3-1 – Create and compare example model versions A -> B -> C to see if the security controls either grow in numbers to address vulnerabilities or decrease over time, reducing overhead. I.e., are the controls more focused and lean instead of numerous and bulky while minimizing entry points and interfaces of failure?
- RT3-2 – Review the model to see if components or interfaces trend toward consuming the most time (consistently problematic with regards to achieving ATO or require the most maintenance [prone to restarts, crashes, failures])
- RT3-3 - Compare a system's security controls' beginning and current state to see if upgrades and patches are current and if their capabilities improve reliability and maintainability.
- RT3-4 – Compare the time from version A through N to achieve ATO. Examine if reaccreditation of the system takes less time as the model matures. Hypothesize if it is possible to use the accredited model for a new program and reduce cost.

This research will contribute to better understanding and measuring MBSE's impact on the ATO. It will also help determine whether integrating MBSE as part of the ATO process can realize a return on investment and provide foundational support to aid program stakeholders. The chapters that follow present the results of this research agenda, the conclusions derived, and recommendations for future work.

# CHAPTER 3 – MODELING THE AUTHORIZATION TO OPERATE PROCESS AND THE EXAMPLE INFORMATION SYSTEM

Chapter Three presents a comprehensive analysis of the ATO process architecture and demonstrates how USG Information Systems can be effectively represented through MBSE modeling techniques. It offers a general background on this model's development and the objectives for this dissertation.

This chapter first presents the methods by which the ATO was represented in an MBSE environment, to assess the costs and benefits of a model-enabled ATO. The dissertation then presents the development of the model, representing an example of USG IS, and the methods by which these systems' characteristics will be compared to the baseline document-centric ATO's characteristics.

All models were developed using an industry-standard tool and language.

| | |
|---|---|
| *Modeling Tools* | Dassault Systemes'©<br>• Cameo Magic Systems of Systems Architect<br>• Cameo Systems Modeler |
| *Modeling Language* | Systems Modeling Language (SysML) |

SysML is a general-purpose system architecture modeling language for Systems Engineering applications (SysML.org, 2024)

## 3.1 SYSTEMS MODELING LANGUAGE (SysML) MODEL OF THE ATO

MBSE has the potential to bring a host of benefits to the ATO process. One key benefit is it promotes collaboration by centralizing the data. In doing so, this provides visibility into the most current version of the data and model. Stakeholders, decision-makers, and all parties involved with the program have access to the latest information, which provides insight into the program's current state. The information provided by integrating MBSE promotes better decision-making.

MBSE is an innovative method that involves designing, analyzing, and documenting complex systems using a model-based approach. MBSE enables a more collaborative and integrated approach to engineering, where system models become the primary means of communication between all stakeholders in a project. MBSE promotes clarity and collaboration in the design process, increases the accuracy of specifications, improves traceability, faster system development, and reduces overall costs. Since models visually convey a system's design, they significantly improve communication and facilitate a better understanding of the system's requirements. Innovation is also enhanced when all stakeholders can view the same model, making the identification of issues or enhancements early in the design process more manageable.

Both Figure 11, and Figure 12 present example diagrams from the SysML model of the ATO process, presented first at a high level of abstraction.

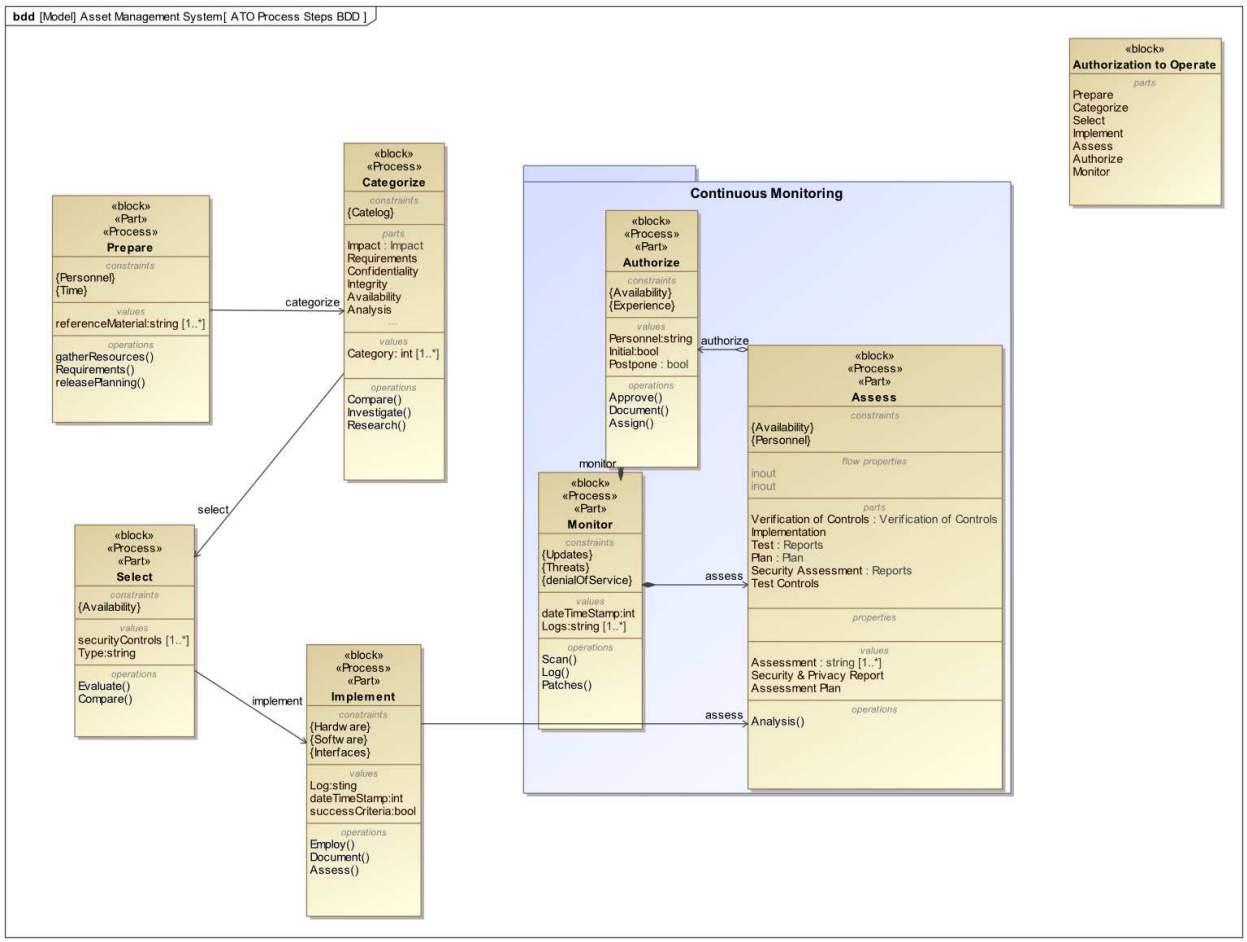



*Figure 11 - ATO Process Steps Block Definition Diagram*

Within each structural element presented in Figure 5, there is a comprehensive model of the steps and data required to execute the ATO. As an example, in Figure 12 the "Categorize" step is the first place where high-level characteristics of the system are identified ("Categorized"). An Internal Block Diagram (IBD, as in Figure 12) represents data flow and outputs from this step of the ATO process. A similar level of modeling detail is present throughout the model to represent the architecture and data associated with the ATO process. The model in its entirety is available for distribution at www.engr.colostate.edu/se

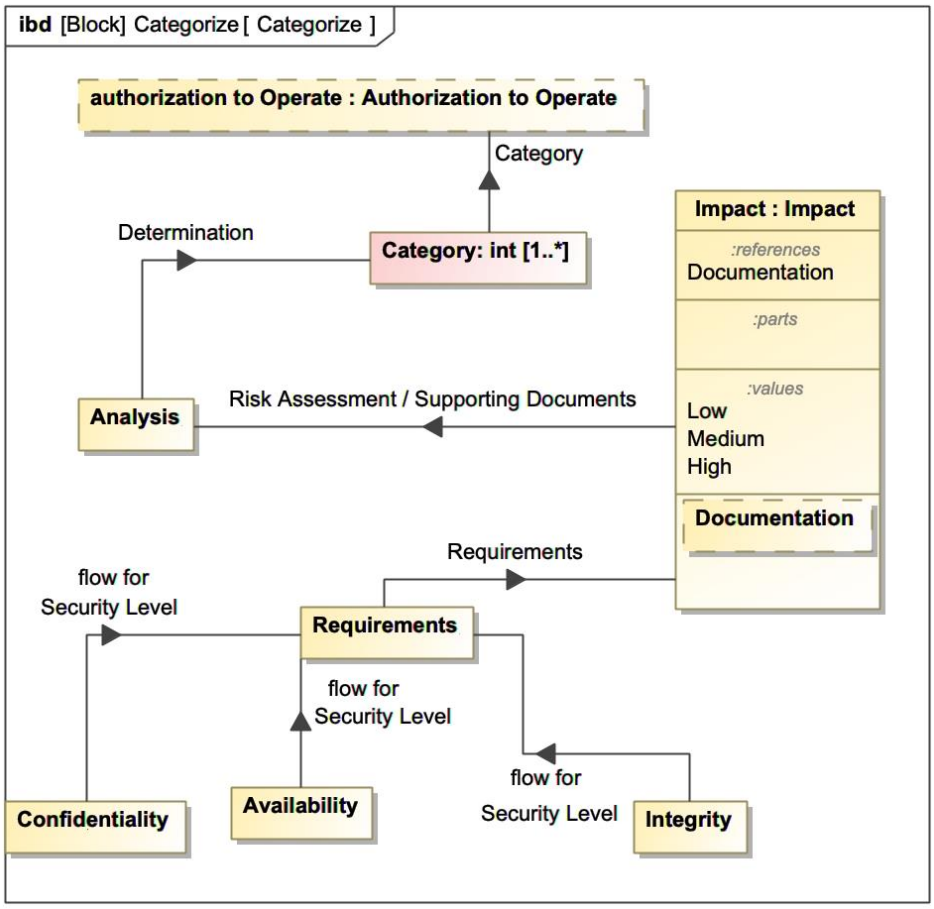



*Figure 12 – Internal Block Definition of the "Categorize" Step in the ATO*

Furthermore, a Block Definition Diagram (BDD) was developed to show the elements of the ATO and their relationship to the IS under development. Figure 13 illustrates the actors involved in conducting the ATO process represented as blocks, specifically the decision authority and government and contractor security officers. At a high level, the diagram shows previously accredited commercial software available for system development and some of the entities needed to support the IS construction. These blocks have been developed with the intention of reusability for other systems that may be needed stemming from similar requirements.

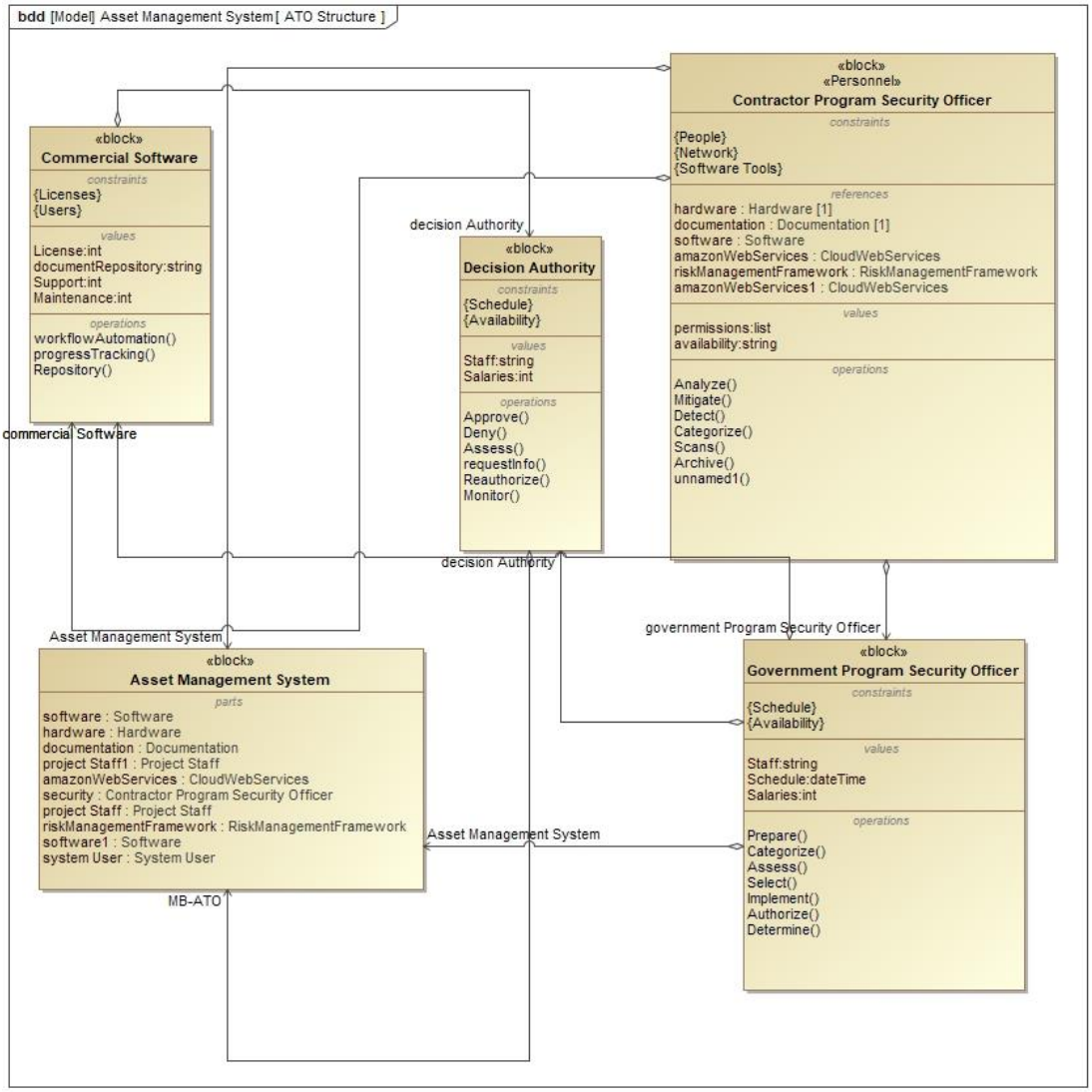



*Figure 13 - Block Definition Diagram of the ATO and IS*

To visualize the organization of a typical Government IS development project, the BDD below represents the main actors, represented as packages and decomposed into blocks for each entity.

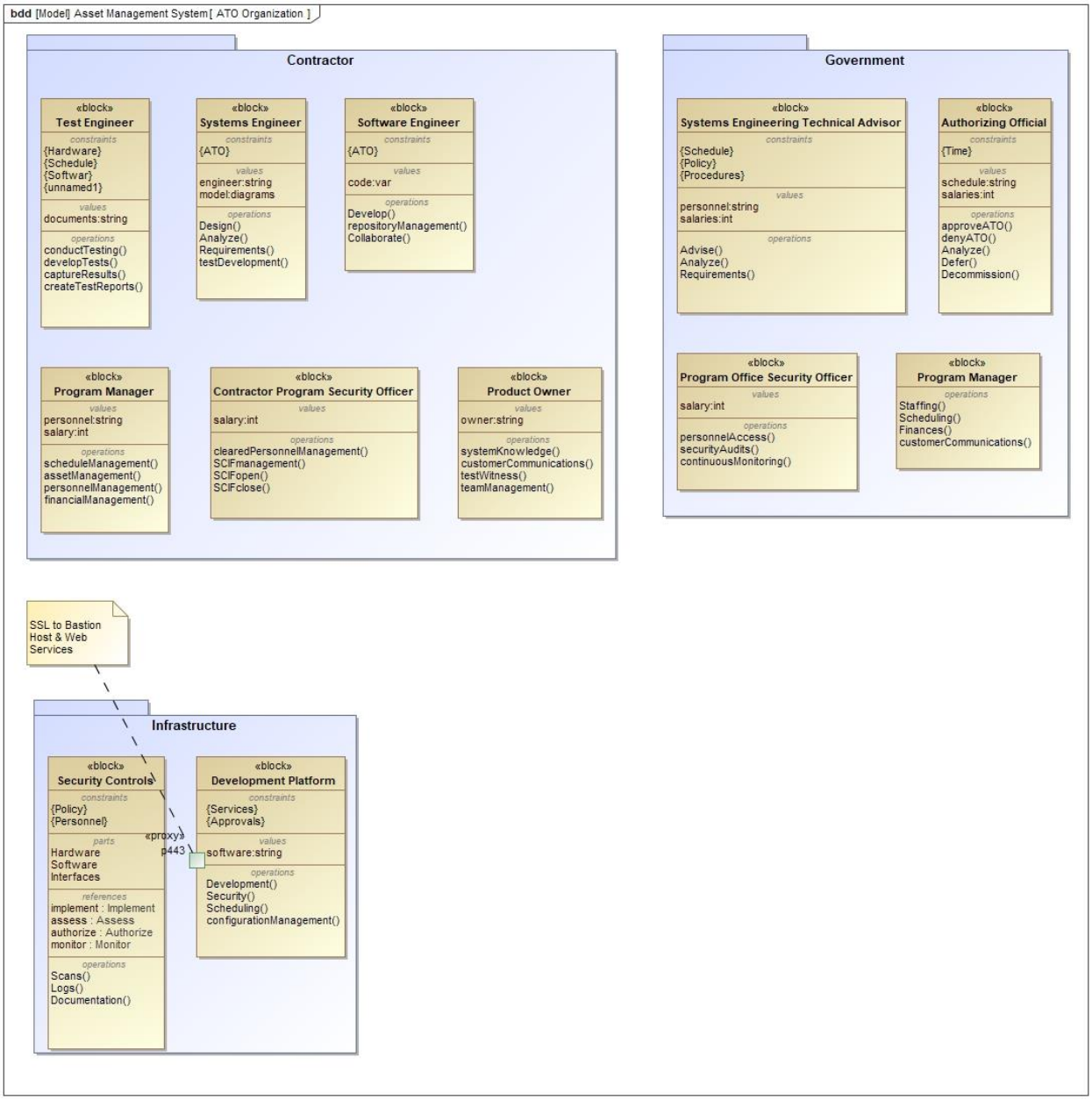



*Figure 14 - Asset Management System and ATO Entities BDD*

## 3.2 SYSTEMS MODELING LANGUAGE (SysML) MODEL OF AN INFORMATION SYSTEM

To realize the benefits of a model-enabled ATO process, there must be a model-based representation of a USG IS that is subject to accreditation. For this research, a representative

(fictionalized and unclassified) model of an Asset Management System[3] was developed. This model represents what an MBSE artifact would look like if developed using a modern, web-based, cloud-native software and hardware solution. At a high level of abstraction, the Use Case Diagram shows communications among system transactions (*Use Cases*) and external users (*Actors*) in the context of a system boundary (*Subject*; notation: rectangle). Actors may represent "wetware" (defined as persons, organizations, facilities), software systems, or hardware systems. Defining relationships between the system Subject and the system Actors is an effective informal way to define the system scope [43].

[3] This fictional system is created as a reference Information System (IS) which will sometime referred to as simply an "IS" in the documents to communicate its commonality to representative systems in the Government domain.

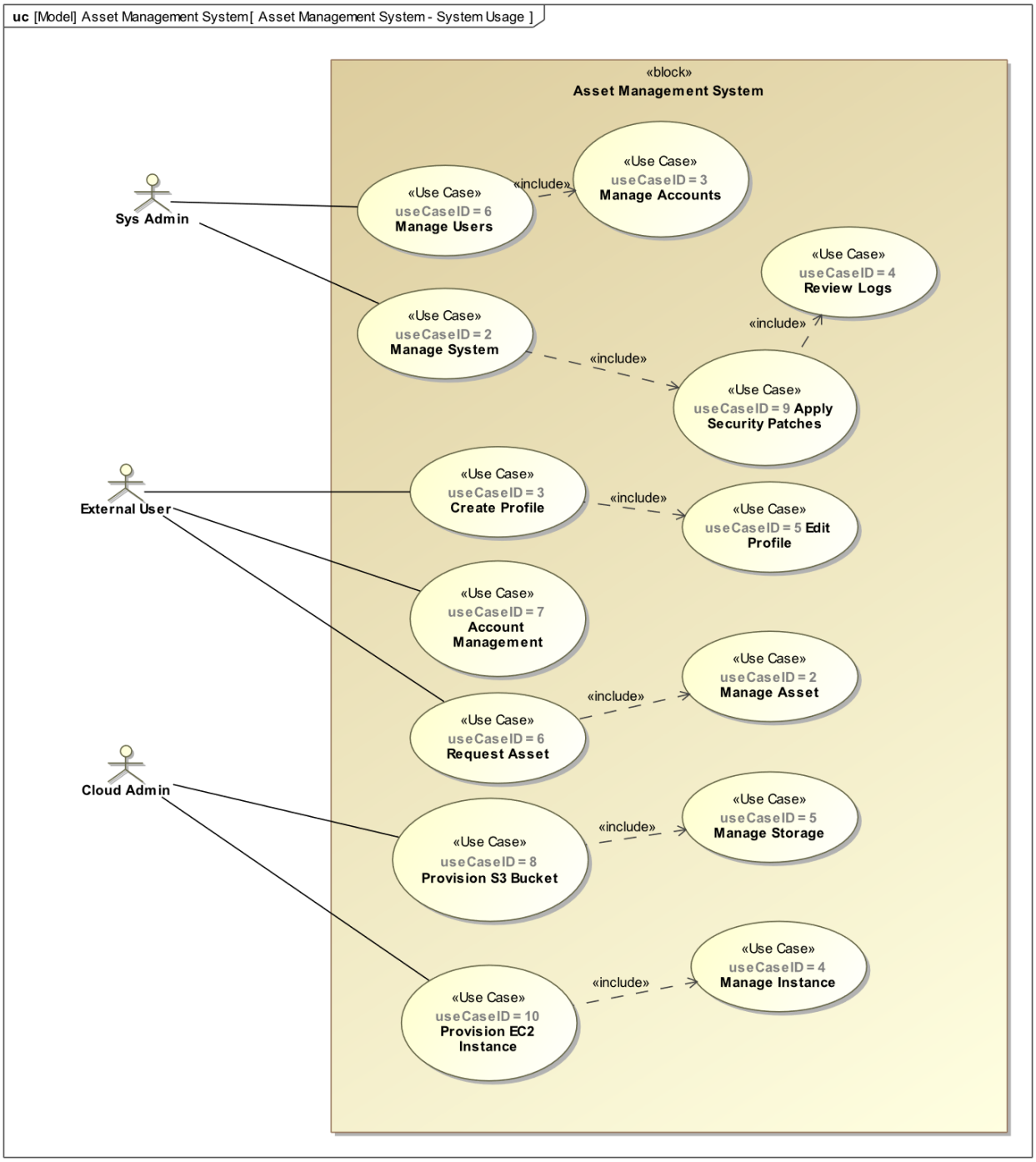



*Figure 15 - Asset Management System Use Case*

This model of a USG IS is based on the specifications and guidance from the DoD Cloud Strategy [44]. This fictitious system uses the resources available as approved by the USG for cloud application development. Most of these tools are commercially available and have undergone a thorough security analysis before being adopted and integrated into the USG's cloud infrastructure.

The accreditation process for the commercially available tools, even from trusted vendors, must also follow the ATO process. Therefore, there is often a lag between the development of commercially available tools and their accreditation and availability in the USG cloud. As such, this IS (a fictitious Asset Management System) accurately represents the architecture and functions found in current cloud computing USG systems. The system is presented as a BDD. Key components of this structure/data view of the IS include key personnel, hardware, software, and documentation as examples. Generally, a System Element Specification, as shown in Appendix A [32] can accompany the BDD to provide additional information specific to each block. An example is given for the Project Staff Block of the IS in Table 9 - System Element Specification - Project Staff located in Appendix A – System Element . Embedded within each of these models are operational/functional diagrams and a requirements model [45] diagrams for the IS.

When decomposed, this Use Case expands to show a copious amount of personnel, software and hardware assets, documentation, development and authorization processes, and third-party tools (some open source) used in the system's development. It becomes increasingly complex quickly, straining the traditional approach to keep pace with the rapid development of modern practice.

Using this model of the IS, there can now be a representative of the processes that guide similar cloud-based systems through the ATO. This creates a baseline for measuring the effects of architecture reuse, the capability to reduce inconsistencies, and an opportunity to manipulate the impacts of security controls.

The USG IS is illustrated as a BDD in Figure 16. Key components of this structure/data view of the IS include key personnel, hardware, software, and documentation. Generally, a System Element Specification [32] can accompany the BDD to provide additional information specific to

each block. An example is given for the Project Staff Block of the IS in Table 1 of the Appendix A – System Element Specification. Within each of these models are operational/functional models and requirements models for the IS. The model in its entirety is available for distribution at www.engr.colostate.edu/se.

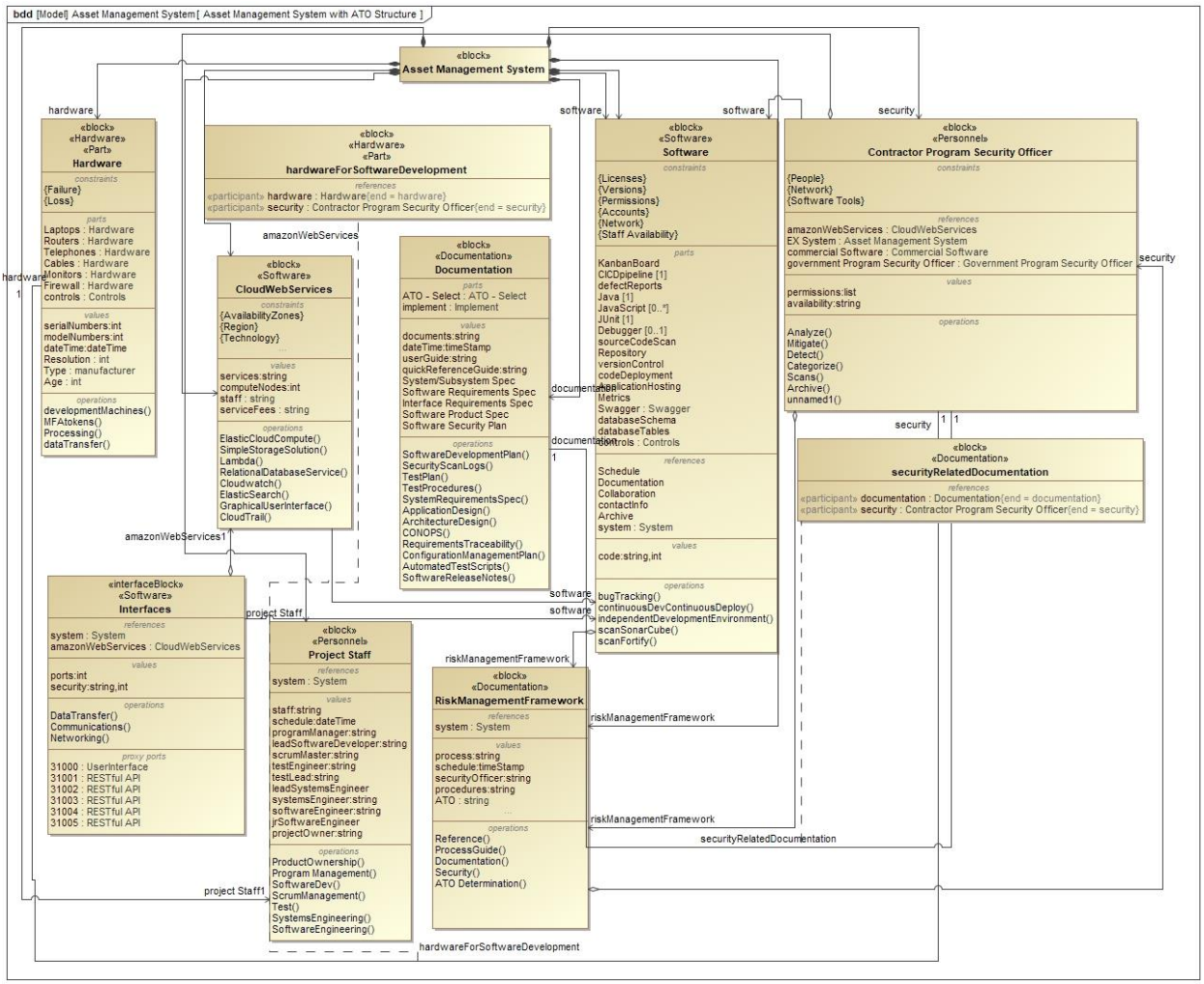



*Figure 16 - USG IS Model in the form of a SysML BDD diagram*

The following Content Diagram illustrates the various model elements of the Asset Management system. Each model element offers specific insights into the structure and activities of the Asset Management system, as well as its interactions with the ATO.

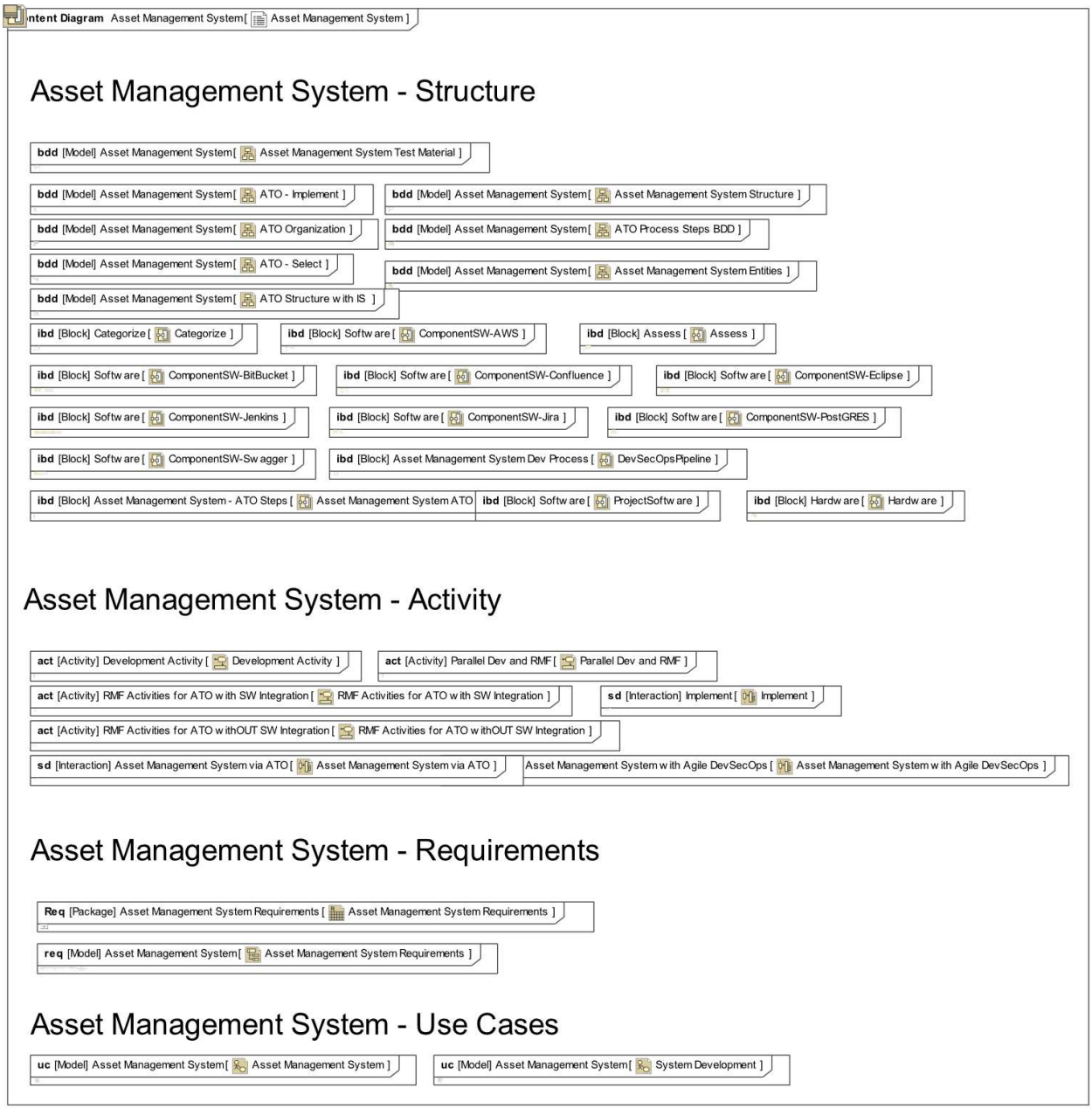



*Figure 17 - Asset Management System Content Diagram*

## 3.3 DISCUSSION

This chapter outlined the construction and key attributes of the SysML models for the ATO and USG IS (specifically the notional Asset Management System). With these models established, this research can now evaluate them, their interactions, and their artifacts in terms of their costs and benefits for facilitating an MBSE-enabled ATO process. Notably, several advantages over the traditional DB-ATO approach become apparent. One significant advantage is the ability to quickly

visualize the system and its components through a centralized repository, eliminating the need to sift through thousands of pages and cross-check references from various documents. This framework allows for a thorough assessment of the application of MBSE within a system that shares architectural and design elements common to Government ISs. Additionally, the model of the ATO serves as a valuable tool for analyzing the ATO process itself.

This process entails some initial overhead and necessitates a level of modeling expertise. It is crucial to consider software licensing, and the technical skills required for effective implementation. These requisites will be explored in greater detail later in this dissertation, along with an analysis of the costs and considerations involved in transitioning to a model-based approach for system accreditation. In this dissertation, all modeling knowledge and experience are derived from courses offered at CSU. The university generously provides access to modeling software.

# CHAPTER 4 – ASSESSING THE MBSE-ENABLED ATO TO REDUCE INCONSISTENCIES

This chapter summarizes the research results to reduce document-centric systems engineering process inconsistencies in documentation, draws conclusions, and provides recommendations for future research and investigations.

## 4.1 DESCRIPTION OF RESEARCH QUESTION ONE (1)

Inconsistencies in documentation and version control (Configuration Management) can lead to confusion and increase the costs of a program. This RQ seeks to understand whether an MBSE model and tool can effectively provide error correction in real or near real-time and can mitigate areas prone to confusion due to inconsistencies. This will promote efficient development and deployment. Table 2 illustrates the types of errors commonly present in ATO technical documentation. These are repeated here in bullet form.

- Bad references
- Vagueness
- Bad mapping to requirements

These types of errors are difficult to correct in a document-centric ATO process. Modern MBSE tools and models have means to detect and correct these types of errors. RQ 1 seeks to measure and validate the effectiveness of these types of tools in the ATO document generation process.

Research Question One (1) - *Does an MBSE model in the development lifecycle and ATO process reduce two typical inconsistencies in traditional document-centric systems engineering?*

## 4.2 INTRODUCTION

Inconsistencies in systems engineering and systems development projects pose a risk to security, timeliness, and cost to systems. They often lead to cost overruns and undesirable overhead to identify and mitigate the problems. Inconsistencies lead to confusion, errors, bugs, and potentially poor decision-making. In the most extreme cases, inconsistencies could lead to loss of life if improper validation and verification is not appropriately performed to defense systems.

### 4.2.1 CHALLENGES WITH REQUIREMENTS TRACEABILITY AND CONSISTENCY IN THE ATO

One of the most common problems the author has encountered in practice with the baseline document-centric ATO process is poor traceability and consistency of requirements. In the document-based ATO (DB-ATO) process, requirements management is most often performed manually using a Requirements Traceability Matrix (RTM), commonly in the form of a spreadsheet. An RTM maps requirements to the validating test procedures that demonstrate that the requirement has been satisfied. When following a traditional, document-based approach, a variety of errors can lead to poor requirements traceability and consistency. Errors that the authors have found in a sample of ATO RTMs and associated documents include:

- Inaccurate or incomplete system architecture illustrations,
- Broken traceability between requirements and design documents,
- Broken traceability between requirements and test documents,
- Inconsistent terminology between documents,
- Conflicting risk assessments,
- Inconsistent documentation of system boundaries and interfaces,
- Invalid or “broken” hyperlinks because of:

- Typographical errors in the hyperlink,
- The document has been moved to another location,
- The document has been renamed,
- The document has been deleted,
- Noncompliance with documented industry standards.

Figure 18 - Example Test Procedure Test Step for Requirements Verification illustrates a subset of these errors in the form of an excerpt of a Software Test Procedure document representative of real-world ATO documentation. As part of this step of the ATO process, for each test (Test Step 44, see label (1), the engineer assembles a list of references to internal requirements derived from customer requirements that a multi-step test procedure would validate. These requirements should be traced back to the source requirement document. As illustrated in Figure 18, this single test "step" is responsible for demonstrating that the system meets eleven (11) derived requirements[4]. Assigning these requirements to the test steps is a long, laborious, manual process where each requirement description must be mapped to the test step and then up to the customer-provided system requirements document. In this case, these requirements were initially assigned by the test procedure author, a software tester with expertise in test procedure development. This person is often not a systems engineering expert, and they may have little to no experience with requirements engineering procedures and structures. As the test procedure is executed (see label (2)), each requirement (see Figure 18, label (3)) is manually read and evaluated to ensure the step has addressed it correctly. This highly manual process is the most common source of error and inconsistency. The derived requirement is commonly mapped to the wrong test

[4] It is also not good requirements testing practice to assert that this single test is traceable to and responsible for verifying so many requirements, which include functional and non-functional requirements derived from a variety of stakeholders and domains (ITS, MSQ, IGR, UPT).

step, or the mapping from the derived requirement to the customer's system requirements document is incorrect. There are often typographical errors that happen when adding the derived requirement document (in this example, the prefix "SYS" was mistakenly used to indicate system requirements instead of the proper prefix "ITS," which references IT systems). In this case, requirements data was scattered across different tools (MS Word, Excel), versions were often not centralized and updated, and the result was a common set of inconsistencies and errors.

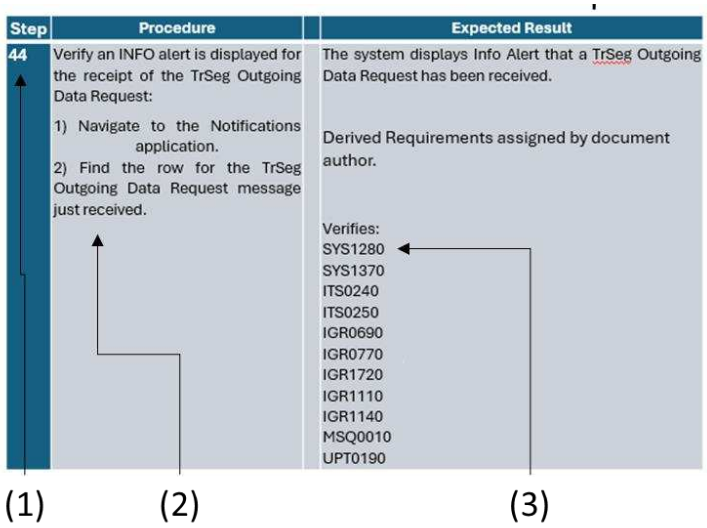
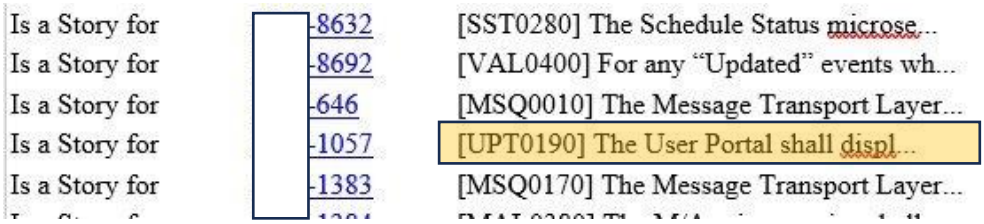



*Figure 18 - Example Test Procedure Test Step for Requirements Verification*

The figures display continues to illustrate consistencies that were identified in a real-world USG IS development program. The formatting is an obvious inconsistency that further exemplifies a lack of proper development processes and configuration management. However, more importantly there is a redundancy of work when examining the documentation more closely. The figure below shows a story that was created which includes a verification of the requirement UPT0190 (a part of the story ID has been obfuscated to remove any association to the program).



| | | |
|---|---|---|
| Is a Story for | -8632 | [SST0280] The Schedule Status microse... |
| Is a Story for | -8692 | [VAL0400] For any "Updated" events wh... |
| Is a Story for | -646 | [MSQ0010] The Message Transport Layer... |
| Is a Story for | -1057 | [UPT0190] The User Portal shall displ... |
| Is a Story for | -1383 | [MSQ0170] The Message Transport Layer... |

*Figure 19 - Example of redundant verification of requirement*

Referring back to Figure 18, UPT0190 was verified as tested by Step 44 of the respective test procedure, which is not associated with the Story referred to in Figure 19. It is not beneficial to verify a requirement more than once, nor is it recommended or necessary. Furthermore, the requirement was tested yet again in a totally different test procedure as shown in Figure 20. This increases confusion, time to deploy the system, and cost as this redundant work will require time to resolve.

| | | | | | |
|---|---|---|---|---|---|
| 7 | Verify the Schedule Data message is received in Ingress.<br><br>View the GUI Notifications for Info alert for receipt of the Schedule Data message. | | The GUI Notifications shows Info alert for receipt of the Schedule Data Message.<br><br>Verifies UPT0190 | | |
| 8 | Verify the Message Transport Layer notifies the M/A microservice of an Alert.<br><br>View the M/A microservice logs for message from Message Transport Layer.<br><br>In Git Bash:<br>Type kubectl logs -f {M/A microservice] | | M/A microservice is notified of alert. | | |

*Figure 20 - Redundant requirement verification - test procedure*

### 4.2.2 CHALLENGES IN MAINTAINING CONSISTENCY BETWEEN SECURITY CONTROLS AND ATO DOCUMENTATION

The author has identified a common issue in the baseline document-centric ATO process: inconsistent reuse of security controls and documentation, particularly in the context of cloud-based U.S. Government Information Systems. This research aims to address the Department of Defense (DoD) Cloud Strategy, which outlines the responsibilities for processing and disseminating information related to military operations, intelligence gathering, and other associated activities. Since this strategy mandates a shift to cloud-based information systems, many of these web-based systems exhibit similar architectural characteristics. Consequently, these cloud architectures present opportunities to implement and reuse security architectures, software, security controls, and analyses, ultimately resulting in anticipated cost savings and enhanced security.

In the DB-ATO process, the reuse of security architectures and controls is typically handled manually. While the reuse of security controls can be advantageous—allowing the development and documentation of a set of best-practice controls to be applied across various projects—the manual nature of this process also raises the risk of errors. Specifically, mistakes can occur when components are carelessly copied and pasted from one document to another, leading to the misapplication of security controls. For example, the author has encountered a situation where a physical security control (such as gates and locks) was referenced in a textual description of a virtualized system's architecture, rendering it out of context and entirely irrelevant.

Such copy-and-paste errors are often driven by the urgency for cost savings, a lack of cybersecurity expertise among ATO process managers, and tight project deadlines.

The NIST SP 800-53 standard has undergone five revisions, this dissertation uses the most recent Revision five (5) and includes over 1,000 security controls. This extensive catalog provides federal agencies with recommended security and privacy measures for protecting information systems against potential security threats and cyber-attacks. However, the vast number and complexity of the controls may lead to long-term issues when employing a copy-and-paste strategy. The various families of security controls underscore the necessity of having experienced security personnel involved in the development team. The NIST 800-53 Rev. 5 Control Families are:

AC - Access Control

- The AC Control Family consists of security requirements detailing system logging. This includes who has access to what assets and reporting capabilities like account management, system privileges, and remote access logging to determine when users can access the system and their level of access.

AU - Audit and Accountability

- The AU control family comprises security controls related to an organization's audit capabilities. This includes audit policies and procedures, audit logging, audit report generation, and protection of audit information.

AT - Awareness and Training

- The control sets in the AT Control Family are specific to your security training and procedures, including security training records.

CM - Configuration Management

- CM controls are specific to an organization's configuration management policies. These include a baseline configuration that will operate as the basis for future builds or changes

to information systems, information system component inventories, and a security impact analysis control.

CP - Contingency Planning

- The CP control family includes controls specific to an organization's contingency plan in case a cybersecurity event should occur. These include controls like contingency plan testing, updating, training, backups, and system reconstitution.

IA - Identification and Authentication

- IA controls are specific to an organization's identification and authentication policies. This includes the identification and authentication of organizational and non-organizational users and the management of those systems.

IR - Incident Response

- IR controls are specific to an organization’s incident response policies and procedures. This includes incident response training, testing, monitoring, reporting, and response plans.

MA - Maintenance

- The MA controls in NIST 800-53 revision five detail requirements for maintaining organizational systems and the tools used.

MP - Media Protection

- The Media Protection control family includes controls specific to access, marking, storage, transport policies, sanitization, and defined organizational media use.

PS - Personnel Security

- PS controls relate to how an organization protects its personnel through position risk, personnel screening, termination, transfers, sanctions, and access agreements.

PE - Physical and Environmental Protection

- The Physical and Environmental Protection control family is implemented to protect systems, buildings, and supporting infrastructure against physical threats. These controls include physical access authorizations, monitoring, visitor records, emergency shutoff, power, lighting, fire protection, and water damage protection.

PL - Planning

- The NIST SP 800-53 control PL family is specific to an organization's security planning policies and must address the purpose, scope, roles, responsibilities, management commitment, coordination among entities, and organizational compliance.

PM - Program Management

- The PM control family is specific to who manages your cybersecurity program and how it operates. This includes but is not limited to, a critical infrastructure plan, information security program plan, plan of action milestones and processes, risk management strategy, and enterprise architecture.

RA - Risk Assessment

- The RA control family relates to an organization’s risk assessment policies and vulnerability scanning capabilities.

CA - Security Assessment and Authorization

- The Security Assessment and Authorization control family includes controls that supplement the execution of cybersecurity assessments, authorizations, continuous monitoring, plan of actions and milestones, and system interconnections.

SC - System and Communications Protection

- The SC control family is responsible for systems and communications protection procedures. This includes boundary protection, protection of information at rest,

collaborative computing devices, cryptographic protection, denial of service protection, and many others.

SI - System and Information Integrity

- The SI control family correlates to controls that protect the system and information integrity. This control family includes NIST SI 7, which involves flaw remediation, malicious code protection, information system monitoring, security alerts, software, firmware integrity, and spam protection.

SA - System and Services Acquisition

- The SA control family correlates with controls that protect allocated resources and an organization's system development life cycle. This includes information system documentation controls, development configuration management controls, and developer security testing and evaluation controls [46].

## 4.3 METHODS

To assess the costs and benefits of the MBSE-enabled ATO process concerning inconsistencies, this dissertation will utilize the SysML model of the ATO presented in paragraph 3.1 and the SysML model of the USG IS outlined in paragraph 3.2. These resources collectively facilitate an evaluation of the effectiveness of the MBSE-enabled ATO as applied to the USG IS, particularly in its ability to mitigate inconsistencies in ATO documentation and requirements. This study has adopted a structured approach to requirements modeling to achieve these benefits as well as leveraging MBSE practices and tools to centralize the data.

### 4.3.1 MODEL-BASED REQUIREMENTS

A set of requirements models is included in the SysML requirements diagram for the USG IS. For this system, this research has adopted a structured approach to model-based requirements

development [47] [48] [49]. Structured requirements follow the format of structured "shall"[5] statements:

The **[Who]** shall **[What] [How Well]**

under **[Condition]**.

Each of the four bold attributes are described as follows:

- [Who]: Defines a subject term specified by an agent or user role that provides a capability or performs a function.
- [What]: Refers to an action verb term specified by a required functionality or characteristic.
- [How Well]: Indicates a comparison factor specified by constraints that can be applied to restrict the implementation of a required functionality or a design characteristic.
- [Condition]: Describes the measurable qualitative or quantitative terms specified by characteristics such as an operational scenario, environmental condition, or a cause that is stipulated for a requirement. [45]

For instance, "The Test Engineer shall test the Asset Management System to ensure it maintains 99.9% availability under normal operating conditions." When implemented in SysML, structured model-based requirements facilitate the parametric and object-oriented connection of requirements to the relevant components of the system model, as illustrated in Figure 21. In this example, the "**who**" refers to the Test Engineer mentioned in the Project Staff block, while the System Block represents the Asset Management System. The "**what**" pertains to the test of availability, which is satisfied by the Cloud Web Services Block and the Database Block. The "**how well**" is derived from a requirement established by the system owner or stakeholders, in this case, the availability target of 99.9%. Finally, the "**condition**" is dictated by normal operating

[5] In the requirements domain, "shall" signifies a mandatory requirement, while "will" statements are considered optional but desired.

conditions. It's important to note that "normal operating conditions" can be somewhat ambiguous; stakeholders often define this intentionally to allow for flexibility and interpretation. This is illustrated in Figure 21.

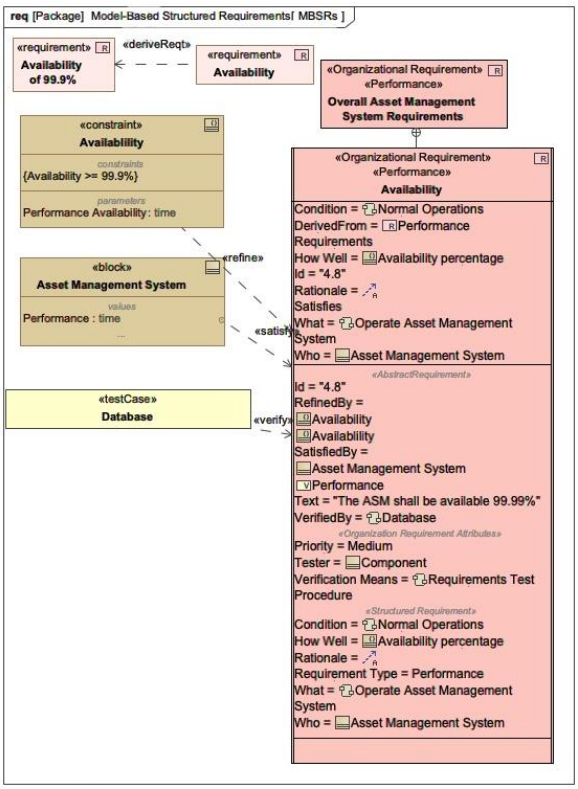



*Figure 21 - Structured Requirements Elements (System, Database, Requirement Text)*

In summary, the model-based structured requirements approach works with the tools and languages of MBSE to enable additional traceability, error detection/identification, and consistency checking. The next task is then to assess the utility and function of this approach to realize these benefits within the context of the model-driven ATO process.

### 4.3.2 DATA CENTRALIZATION

MBSE centralizes all the information (this includes physical attributes, interfaces, external dependencies and other valuable information) about the system in a model, often called the "single source of truth." The model supports the system's entire life cycle, from requirements documentation to validation and verification exercises to maintenance and training purposes.

Stakeholders, like decision-makers, suppliers, and the development teams, can access the model from different views and levels of detail to access data according to their needs. At the same time, the consistency of the information is guaranteed to the level at which the developers are reliable. The MBSE approach improves quality and communication, de-risks the program, and significantly reduces development costs and time. Requirements could be validated very early in the program, avoiding ambiguities, erroneous information, and other specification flaws [50].

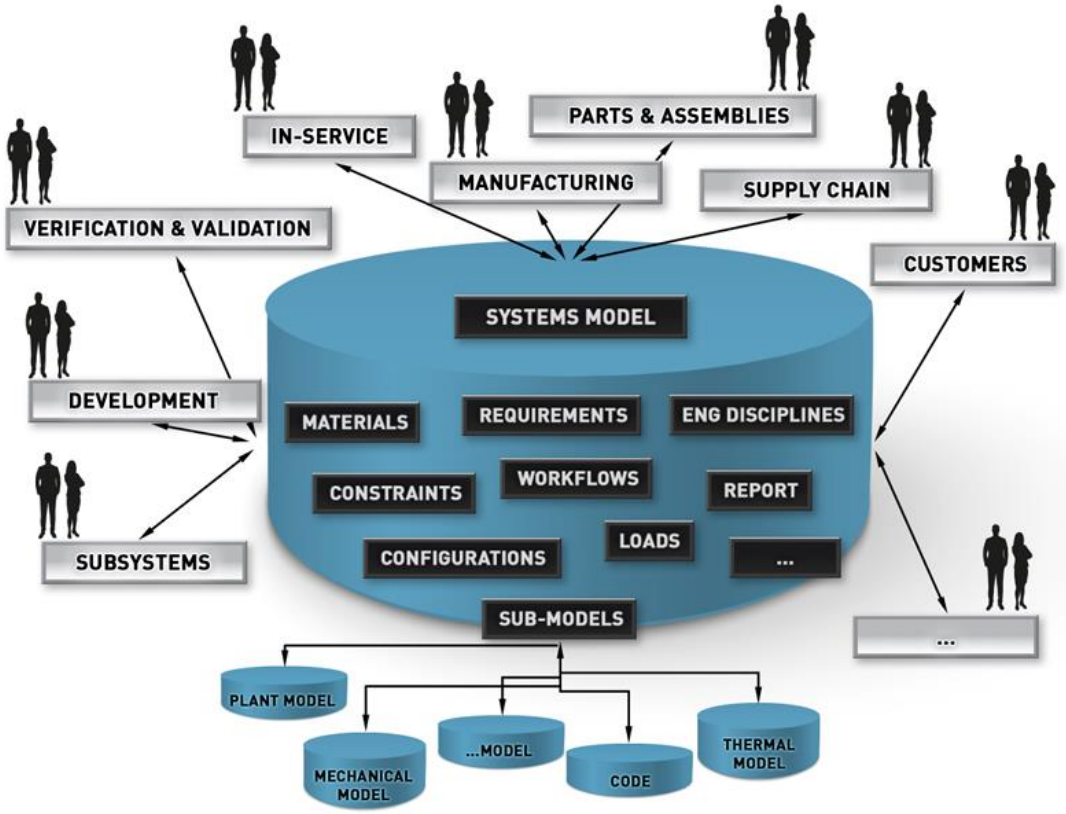



*Figure 22 - Centralized Data Model [51]*

Centralized data is essential to efficiently track changes, desirable or not, on the system. Change Requests (CRs) should be created to address all areas where defects, anomalies, inconsistencies, or new functionality are captured and tracked. By capturing and tracking system changes centrally also reduces the risk of duplication or redundant requests. The CR will serve as a historical record of what was changed, when, by whom, and for what reason. There are many

tools available to capture and track CRs and MBSE can be used to see how functional changes can affect the system prior to integration. Changes to documentation should be tracked similarly since this is where many inconsistencies exist. In this case, templates should be used to promote reuse and reduce inconsistencies for documentation required for ATO and system development (discussed later in section 5.5.3). To further bolster configuration management, modeling tools capture the changes made to the model, which is generally driven by changes in documentation, creating an accurate historical record. Incorporating MBSE as the focal point of the systems engineering process provides content management tools not readily available in DB-ATO. These tools are used to identify incorrect or missing associations, and errors can be corrected before they perpetuate. This is extremely valuable as early detection prevents inconsistencies from propagating throughout the system's development and documentation, saving both time and money.

## 4.4 RESULTS AND DISCUSSION

This section presents the results and evaluation of implementing the proposed model-based ATO process.

### 4.4.1 MODEL-BASED REQUIREMENTS TRACEABILITY THAT VALIDATES THE ATO

The benefits of the model-based approach to ATO are presented first by demonstrating a model-based requirements traceability, testing, and validation approach through model-based development of requirements and model-based test planning using Model-Based Structured Requirements (MBSR) methodology. The set of structured requirements (using the template from Figure 8) is modeled within a group of related requirements. As illustrated in Figure 23, the structured requirements packages can be connected to highlight a set of performance measures that must be satisfied before the Asset Management System can be connected to other relevant Performance Requirements and Test Procedures. Subsequently, the Test Procedures will be used

to verify that each Performance Requirement is associated with at least one specified Test Case. This requirements model demonstrates each direct, and auditable, traceability from each requirement to a specific test.

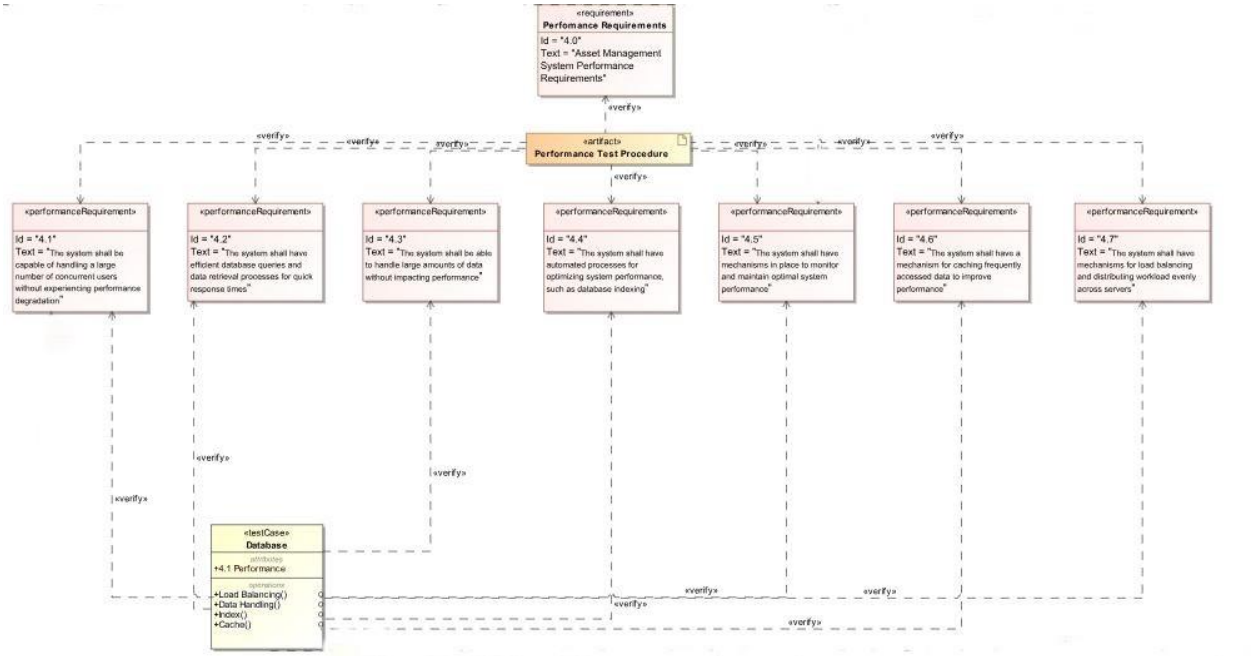



*Figure 23 - Asset Management System - Performance Requirements*

In this example, the majority of the requirements are satisfied by a Database Test Case, represented as a <<Test Case>> element in yellow with specific operations that must be tested to fulfill the requirement. The Test Case includes detailed Test Steps that provide instructions to the operator on how the system is expected to perform under normal operations. This is only a subset of the Requirements. A complete table of example requirements used for this dissertation are available in Appendix D – Asset Management System Example Requirements.

When requirements are entered as model-based structured requirements in modern MBSE tools, their references are automatically checked for accuracy and consistency, and reference material can be inserted as part of the model. This is illustrated in Figure 24 - Asset Management System Test Structure, which presents the Test Procedure Document model (for which an example

of the document was presented in Figure 23 - Asset Management System - Performance Requirements) for the USG IS Asset Management System. Figure 24 illustrates the connectivity between the test cases and how the requirements are satisfied. A direct link to the test document is also embedded in the *Package* element in *Cameo*.

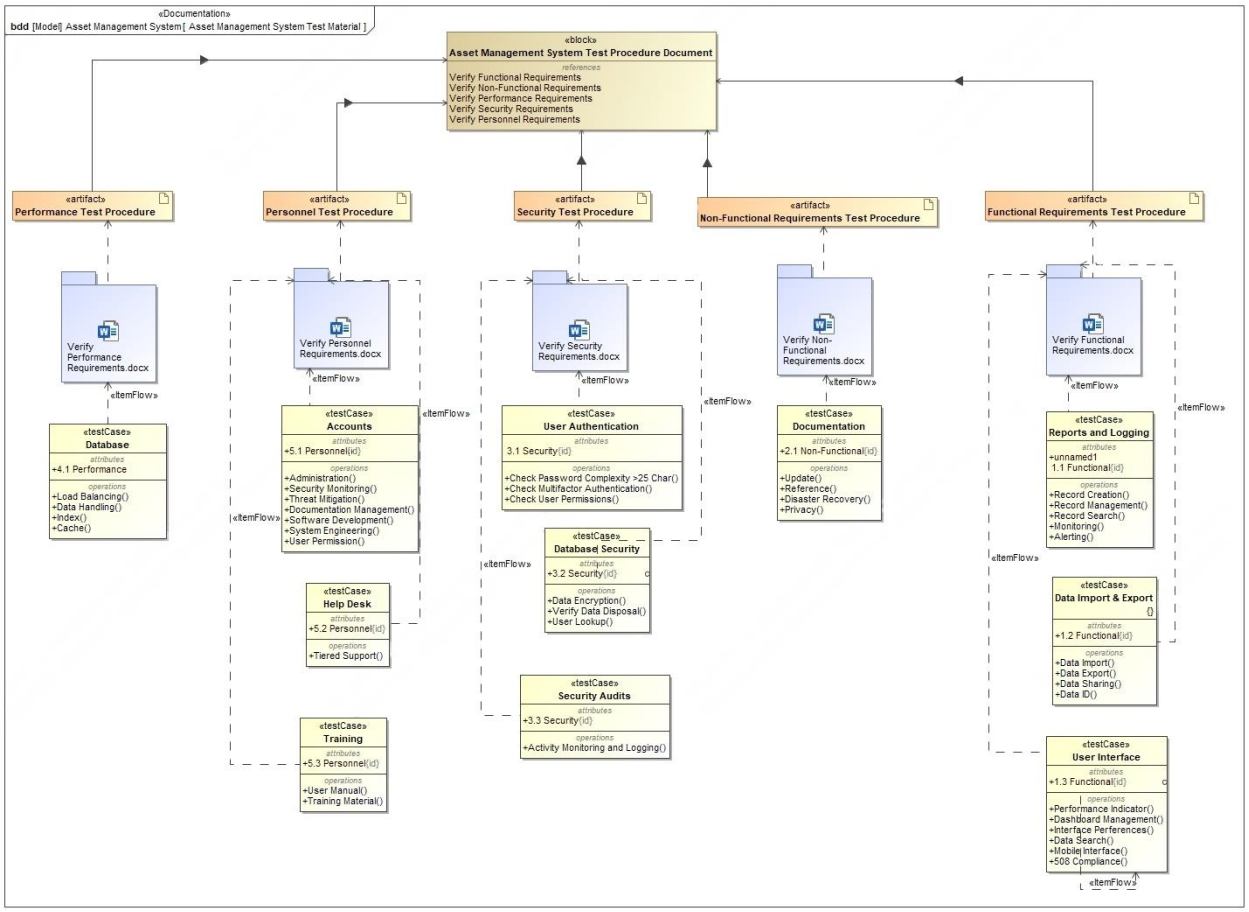



*Figure 24 - Asset Management System Test Structure*

The model-based ATO now allows the user to maintain consistent and up-to-date information in the integrated digital model. Completeness, consistency, traceability, and contradiction checks can be performed.

Once a system has been modeled using a modern MBSE tool, built-in error-checking is intrinsic. By creating relationships and dependencies on requirements, the automatic error-checking prevents the engineer from mapping a requirement to an element that does not exist in

the model. Within the modeling tool, inconsistencies or a lack of requirements and traceability can be identified quickly, visually alerting the engineer that the resource or reference does not exist within the model space.

More quantitatively, the model-based ATO also allows for checking the completeness of the requirements and error checking of requirements. By modeling a requirement statement with its required attributes, engineers and managers can more readily quantify and visualize several indicators of requirement quality and identify gaps/missing information. Different requirements will be incomplete throughout the development, but the eventual goal is to have a complete definition of requirements in the model and the corresponding system design. If various metrics that reflect the model's current state (and system development effort) are tracked, then the evolution of quality and completeness can be rigorously evaluated. Based on the definition of an MBSR above, several metrics can be readily defined and computed:

- Percentage of MBSRs that are complete where completeness is defined as a requirement having nonempty [Who], [What], [How Well], and [Condition] attributes,
- Total number of MBSRs,
- Nonempty attribute MBSRs counts for each of the four structured attributes.

For this IS, a script (written in Groovy, an Apache open-source programming language) is used to compute the nonempty [Who]. The information captured through MBSRs permits additional metric calculations, and the metrics above might be combined with other traditional SysML requirement relationships (e.g., ≪satisfy≫) for a more comprehensive completeness condition [45] [48] [52]. Note that the results for this model show 100% of the requirements allocated.

In summary, the results illustrate that the model-based ATO process promotes the development of MBSRs that take advantage of the intrinsic error, traceability, and completeness checking capabilities of modern MBSE tools that improve the consistency of requirements. By running these checks early in the ATO and system development process, a model-based ATO approach enables the identification of defects early in the system lifecycle, resulting in the potential for higher-quality products and corresponding cost savings [53] [52].

### 4.4.2 MODEL-BASED CONSISTENCY BETWEEN SECURITY CONTROLS AND DOCUMENTATION WITH THE ATO

The benefits of the model-based approach to ATO are presented by demonstrating how the ATO enforces consistency in security controls and security documentation.

In contrast to the DB-ATO process, MBSE is known for centralizing data and artifacts in a common repository to ensure consistency and easy access to systems engineering artifacts, and reusable subsystems such as security controls. The USG IS SysML model, which includes the system's interfaces, designs, architecture, security controls, and documentation, is contained within the tool and therefore accessible to trusted stakeholders.

Within the USG IS model, this author has implemented a variety of security controls that are defined within a SysML package. These controls can then be applied to various and multiple components and processes within the USG IS model. For example, the control that is titled "password management system" is a block that is present and maintained within the "security controls" package. For example, both of these MBSRs ("The password management system shall only allow passwords with a combination of Upper and Lowercase Letters, a number, and at least one symbol," and the MBSR "The password management system shall prohibit users from re-

using the last six previously used passwords") are traceable to the same "password management system" block.

The other benefit of enforcing consistency for security controls across the USG IS model is that the generation of documentation for the ATO process can be performed with automation from a centralized model repository. Referring to Table 3, 41 documents with page counts over 4300 pages were considered candidates for document auto-generation and document replacement directly from the MBSE model of the ATO and USG IS. The documentation highlighted in blue has been identified as candidates for auto-generating directly from the USG IS SysML model. The documentation highlighted in red has been identified as candidates to be directly replaceable by the artifacts of the USG IS SysML model. In this implementation, we have used Velocity Template Language (VTL) to script the auto-generation of documents from the SysML model using the tools available in Cameo. The candidacy for template development is based on the frequency with which these documents are used in all USG ISs [54]. When we model the level of effort required to develop these documents using the metric of the Staff Years of Technical Effort (STE) (commonly used when costing a program, as illustrated in Table 2), the savings in U.S. dollars equate to 31% of the ATO documentation cost. This is analyzed in depth in Chapter 6. Further cost savings can be realized if the documents previously identified in blue text in Table 3 are generated using preexisting templates.

*Table 3 - Common system documents required to perform the ATO*[6]

| ATO Step as Defined in RMF | Non-ATO Documentation – Additional but Recommended – Often Requested | ATO Documentation Package - Required | Common in ATO Package - Additional | Total Amt of Docs |
|---|---|---|---|---|
| **Prepare** | CONOPS<br>Software Development Plan<br>Software Installation Plan<br>Software Transition Plan<br>Operational Concept Description<br>Software Test Plan<br>System Requirements | | Privacy Impact Assessment<br>Privacy Threshold Assessment<br>Incident Response Plan<br>Disaster Recovery Plan<br>ATO Boundary Diagram | 41 |
| **Categorize** | System / Subsystem Design Description | System Definition Document | | |
| **Select** | System / Subsystem Specification<br>Software Requirements Specification<br>Interface Requirements Specification<br>Software Product Specification | System Security Plan | | |
| **Implement** | Software Design Description<br>Interface Design Description<br>Database Design Description<br>Software Test Description<br>Software Test Procedure | Updated System Security Plan<br>Status Report | | |
| **Assess** | Software Test Report<br>Software Version Description<br>Requirements Traceability Matrix | Security Assessment Report<br>Security Assessment Plan | | |
| **Authorize** | Software User Manual<br>Software Center Operator Manual<br>Software Input/Output Manual<br>Computer Operation Manual<br>Computer Programming Manual<br>Firmware Support Manual | POA&M<br>Risk Assessment | | |
| **Monitor** | Continuous Monitoring Plan | Monitor Strategy Document | | |

## 4.5 CONCLUSION

This chapter is written in response to RQ 1, which is: Research Question One (1): *Does an MBSE model in the development lifecycle and ATO process reduce two typical inconsistencies in traditional document-centric systems engineering?*

This research provides evidence to support the idea that the MBSE-enabled ATO does reduce inconsistencies relative to document-centric ATO processes. This research has identified and exemplified key challenges with Requirements Traceability and Consistency and with Consistency of Security Controls and Documentation. To test whether the benefits of MBSE are available within the application of a model-based ATO of a modern, cloud-based USG IS, this research relied on the developed SysML models of both the ATO and an example IS. By

[6] Documents labeled with blue text are amenable to automatic generation from the model, documents labeled red are replaceable by the model artifact itself, documents labeled black are not near-term amenable to automation in this application

leveraging model-based structured requirements, we were able to significantly reduce inconsistencies in requirements traceability. This enhances the overall quality and reliability of the requirements management process, ensuring that all stakeholder needs are accurately captured and addressed.

In addition, the modeling of security controls has proven beneficial in facilitating the reuse and documentation of these controls. Through automated documentation generation from models, we can streamline the process, reduce manual errors, and ensure that all necessary security controls are properly articulated and maintained.

The results of this study indicate that the anticipated advantages of MBSE—such as improved traceability, enhanced documentation, and increased efficiency—are not only theoretical but can be effectively applied within the ATO framework. This opens new avenues for applying MBSE principles to optimize and transform the ATO process, ultimately fostering a more secure and efficient operational environment for U.S. Government systems. The potential for these tools to enhance ATO documentation and approval processes underscores their value in achieving compliance and operational excellence in a rapidly evolving technological landscape.

# CHAPTER 5 – MODELING FOR REUSE

This chapter discusses the application of MBSE reuse of model artifacts and its impact on the ATO process. It also discusses best practices and how reusability affects the cost and schedule of the systems engineering lifecycle.

## 5.1 DESCRIPTION OF RESEARCH QUESTION TWO (2)

Reuse is frequently emphasized in software development projects, particularly in the context of code reuse. This approach often helps to reduce costs while simultaneously enhancing quality. This research posits that the reuse of IT system and ATO process models and artifacts may lead to improvements in the quality and consistency of the ATO. It aims to provide evidence on whether this principle applies to MBSE models and whether reuse can align with budget projections for the program while ensuring continued quality improvement.

Among the most cited advantages of reuse is the potential to save time and money by minimizing, or ideally eliminating, duplication of effort. Software systems have long benefited from reusing code and libraries across different applications, which fosters consistency and standardization among these applications. Similarly, application frameworks and design patterns are reused in a comparable manner.

When MBSE is properly implemented, all stakeholders can reference a single model that encapsulates the system's context and architecture. Instead of each stakeholder duplicating efforts by generating entirely new use cases, block diagrams, or documentation, they can update the architect's model to reflect their concerns and document the outcomes of their analyses. Managing configurations becomes cumbersome when dealing with numerous documents and various

versions of each, so consequently, the potential for cost savings increases with each additional stakeholder who adopts the architecture model as their primary artifact.

Research Question Two (2) – *Does reusing an MBSE model across various similar programs create an improved quality process and, therefore, an improved product in less time? In the context of the ATO process means a more streamlined and lean procedure with improved "ilities" by providing internal consistency and integrity (no loose ends and minimized points and interfaces of failure). How does this affect documentation?*

## 5.2 INTRODUCTION

Documentation across cloud-centric system applications often shares similar characteristics, making it an ideal candidate for templating and reuse. Likewise, MBSE models can be reused due to their comparable architectures, which is a beneficial outcome of the cloud mandate.

Reuse is frequently advocated for in software development, particularly in the context of code reuse. When executed effectively, code reuse helps to lower costs while enhancing quality. This research posits that reusing Information Systems and ATO process models and artifacts may lead to improved quality and consistency within the ATO. This inquiry aims to determine whether the same holds true for MBSE models and whether reuse can support cost projections for the program while simultaneously enhancing quality.

Reusability, bolstered in part by the popularity of Service Oriented Architecture (SOA), is a key component of the value proposition for disciplined system engineering and software development. Reusing architecture, code, documentation, frameworks, and other characteristics is not new. Reuse has been the promised silver bullet in organizations' investments for decades. As software was designed and developed, it was added to corporate and public libraries for reuse [55].

This research uses the object-oriented paradigm to examine software engineering best practices for insights and examples of proper reuse. Improper reuse, such as copy/paste, can be detrimental to development efforts. However, positive reuse – defined as reuse where the benefit outweighs the cost, can have a constructive impact on such projects. Software development projects, large and small, have benefitted by stored and reused classes, libraries, and methods, saving considerable amounts of development time. For example, when a software project begins, it is very common to see statements at the beginning of the code, such as "import," where specific libraries are "imported" into the project. A library is a collection of functions (software capabilities) that can be added to the code so it can be "called" (run) as necessary. There is no reason to continually rewrite code that performs a standard task. With libraries, you can import pre-existing functions and efficiently expand the functionality of the code [56]. Importing a library provides functions that can be used repeatedly without being redeveloped.

This research identifies ATO processes to enable and measure the practice of reusing models and model elements, specifically to allow a more robust, consistent, and streamlined process to achieve ATO. The ATO process encourages the reuse of tools, products, and controls because ATO and the RMF promote a rigorous and repeatable procedure (specifically continuous monitoring and reaccreditation). Therefore, many ATO work products are reused.

Effective reuse for MBSE requires identifying the characteristics of systems engineering that will increase value and improve quality. Some of the constructs used in analyzing the potential for software and product reuse are:

- Patterns. Patterns identify commonly occurring problems and couple them with reusable solutions that are applicable in a stated context. Popularized by software patterns and the "Gang of Four" [57], patterns also apply to systems engineering – as demonstrated

by Cloutier and others. Because of the generic nature of the design pattern and the framework that states the problem, solution, and context, architecture design patterns encode knowledge in an effective format for reuse. Whether drawn from industry practice or simply organizational knowledge, patterns can be encoded in architectural model snippets that can be quickly injected into system models for use in a specific design solution.

- Reference Architectures. Though systems engineering prides itself in its broad applicability to any system, the reality is that we operate within existing organizations tuned to specific markets. While our organizations may demonstrate excellence in systems engineering, they couple systems engineering expertise with the requisite domain knowledge to be successful with specific types of products in specific markets. Therefore, we generally work within the bounds of reference architectures, whether explicitly captured or implicitly maintained in the heads of senior practitioners. Reference architectures are inherently reusable, even more so when captured using model-based practices. The problems are well-bounded, the assumptions generally understood if not documented, and disruptive changes in problem or solution are evident. Capture and reuse of reference architectures – particularly capture in a descriptive architectural model – yields 4 key benefits: aligning the team, starting fast, aligning the resultant systems, and learning.
- Product Line Engineering. Most organizations are actually product line organizations. The question is not whether they release a new version, edition, or generation of a product. The question is how much time passes between releases. When the time is 6 months, 12 months, 2 years or less, product line engineering is viable, and reuse of a corresponding descriptive model brings tremendous rewards. In moving from release to release, many benefits come

from detailed refinements meaning the existing boundaries, assumptions, and design remain valid. Where disruptive changes are introduced – either in customer need, solution approach, or implementation technology – starting with a high-fidelity model-based representation of the previous system with complete traceability enables more effective assessment of impact, identification of risks and concerns, and elicitation of emergent behaviors.

- Analytic Models. The models we use for the analytic dimensions are not new to MBSE. These are models that engineering disciplines have developed over the years, often encoding solid theoretical principles complete with well-understood assumptions and bounds of applicability. As noted earlier, these models are reusable at the system level when done well. The key is good documentation – purpose, limitations, and approach. It is common to hear stories of engineering organizations relying upon 30-year-old Fortran software modules where the original developer has long since retired; no one really understands how to interpret the code, yet everyone still trusts that it is applicable and essential. This perception also holds true when reuse is simply a copy and paste from a source document.

In modern systems and modern SE, reuse is a competitive and organizational necessity. There is neither the time nor the money to custom-design and fabricate system components over and over, and it is not a desirable way of developing systems.

The reuse of analytic models is standard practice today. Reusing descriptive architectural models is coveted but holds the promise of great investment with little return. The wise practitioner instead recognizes that patterns, reference architectures, and product line engineering define a

more appropriate scope for architectural model reuse, delivering tremendous value in the form of lower cost, shorter schedules, and higher quality when implemented correctly [55].

Much of the current ATO process remains manual and document-centric, particularly the substantial reporting and documentation associated with selecting, implementing, and testing the hundreds of security controls required for a baseline USG IS. While authoring the paper, the writer was working through the ATO process on a program with his full-time employer, which was working through aligning accreditation with the desired deployment schedule. Due to the process's scope and the volume of systems requesting approval authorization, the ATO process for an IS can last several years. During this time, vulnerability scanning, and documentation must be continually updated to remain current upon review. Despite the hurdles with the baseline approach, there is still reluctance to embrace MBSE as the solution. A recent co-worker survey showed an agreed acceptance that the ATO process needs to be modernized and more efficient; however, nearly as many of those surveyed were skeptical that a solution is available or mature enough to combat the challenges of the ATO.

The relative improvements available from an MBSE-enabled ATO process can be assessed regarding reusability in three domains: reuse of Security Controls, reuse of Architecture, and reuse of Documentation.

## 5.3 DIFFICULTIES WITH THE COPY/PASTE AND SYSTEM ACCREDITATION

One of the most common problems with the baseline document-centric ATO process is the misuse of the copy/paste reuse method. Copy/paste is a poor example of effective reuse and, in many cases, can create more problems than efficient solutions. One of the most glaring problems of copy/paste is that it simply doesn't scale. A level of abstraction common in code development must be appropriately used to achieve effective and efficient reuse. The goal is to abstract down to

the most fundamental pieces of the system/components / sub-components / code, etc., and reuse those common elements that are unlikely to change repeatedly.

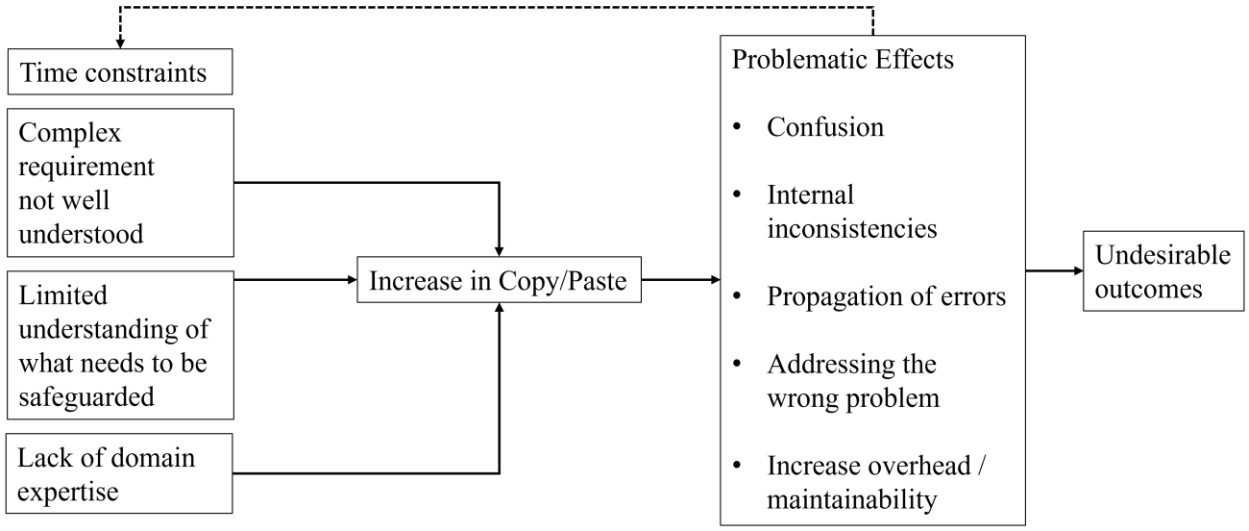



*Figure 25 - Negative Cause and Effect of Copy/Paste [58]*

### 5.3.1 LACK OF DOMAIN AND FUNCTIONAL AWARENESS

Philosophically, copying and pasting is simply a short-term solution that can create longer-term problems, specifically regarding maintainability. “Using copy/paste with code that has yet to be fully debugged can introduce the same error over and over, making final Quality Assurance (QA), as well as long-term maintenance, far more painful than it needs to be [59]”. That is an example of improper reuse. This thought process can be applied to documentation and model elements alike. The critical thinking requirement to build and understand the system's complexity is missing. Therefore, the critical thinking required to problem solve for maintenance, troubleshooting, and debugging is not robust and will require additional time and resources to analyze and problem solve issues. In turn, this has the potential to increase overhead. Overall, this will contribute to negative gains that were assumed to have saved time by copying/pasting in the first place. The process may have resulted in a quicker delivery of an iteration of the system but will also contribute to a lag if / when maintenance or further development is required.

"In requirements engineering, at least 55% of engineers use "copy & paste" to reuse requirements or groups of requirements, and 50% duplicate the full specification. These practices have little added value, as they do not convey a strong understanding of the system, especially concerning its behavior [60]." As was discussed previously, in the DB-ATO process, requirements management is most often performed manually using a Requirements Traceability Matrix (RTM), commonly in the form of a spreadsheet. An RTM maps requirements to the validating test procedures and test cases that demonstrate the requirement has been satisfied. When following a traditional, document-based approach, various errors can lead to poor requirements traceability and consistency, especially when copying and pasting requirements from one system to another. With good intentions but undesirable outcomes, this process is a candidate for documentation templates, not only for reuse, but consistency derived directly from the model. Below is an example of document development based on templates. The elements in blue represent the types of documents that are candidates for templates based on the Systems Engineering "V" model of activities.

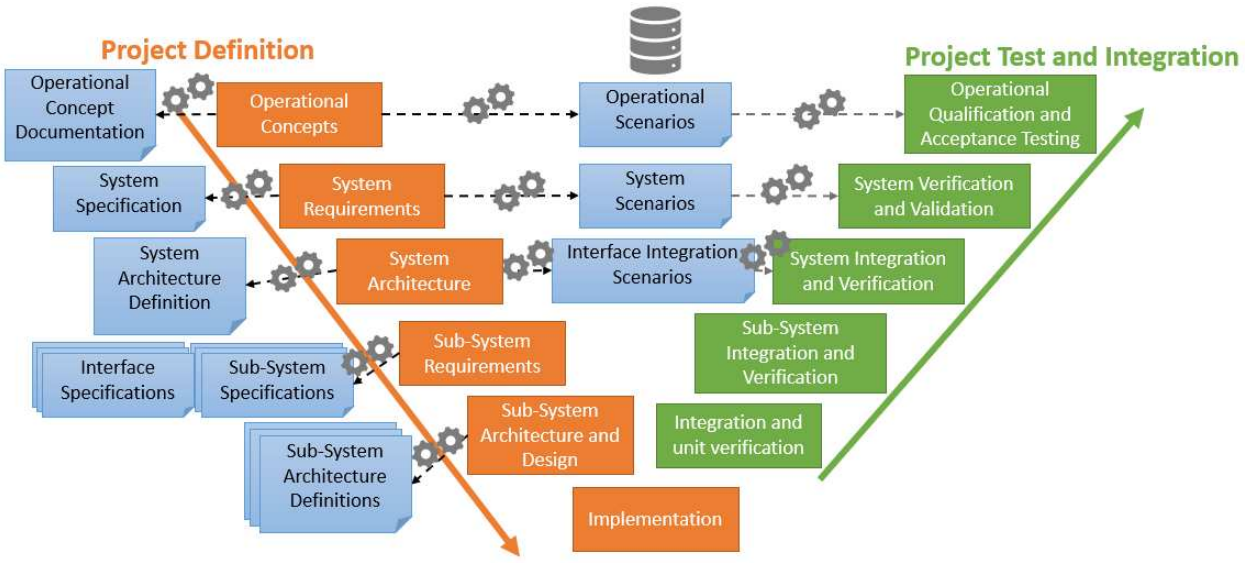



*Figure 26 - System Engineering "V" with Documentation Templates [61]*

### 5.3.2 INCREASED COMPLEXITY THROUGH OBJECT-ORIENTED INHERITANCE

Even when reuse is done correctly, there can quickly, and easily be an increase in the complexity of the source code. This can be measured in terms of Cyclomatic Complexity, which measures the number of linearly independent paths through a program module. Programs with lower Cyclomatic Complexity are easier to understand and less risky to modify [62]. When designing object-oriented software, developers must deal with a conflict between the advantages of inheritance (increased reuse and improved similarity of implementation and problem structure) and disadvantages (increased coupling and complexity). Because of this conflict, developers should limit inheritance and reduce coupling. This can be applied to the sharing of model elements across projects. Though the sharing of the model element from one project to another is encouraged, truly only high-level and common attributes should be reused, and element specifics should be unique (differentiating names, IDs, and values specific to the project). To limit the likelihood of complex reuse, data from Bieman suggests that two design practices:

1. dividing a system into general-purpose and specialized modules and,
2. using multiple inheritance in appropriate situations,

can result in greater reuse with minimal increases in coupling [63].

### 5.3.3 PROBLEMS WITH REUSE OF SECURITY CONTROLS AND DOCUMENTATION WITH THE ATO

In the author's experience with DB-ATO process, the reuse of security architectures and controls is often performed manually. For example, a Systems Architecture Document might be reused many times for different systems to save time and money. However, a common problem with this method is the architecture is poorly updated to meet the new system's design and requirements. Interfaces to external systems are often incorrect or not appropriately secured. Some

artifacts left over from the previous system are irrelevant to the new system. The manual nature of the process lends itself to opportunities to cut corners. The cutting of corners is most obvious and commonly observed, with copy-and-paste additions from one source to another. Security control selection/implementation is based on the system's calculated Confidentiality-Integrity-Availability (CIA) rating. Once the CIA rating is determined, the controls are selected using Table D-1, within Committee on National Security Systems Instruction (CNSSI) 1253 [64]. But, for similar systems, some of these security measures are copied from one system to another. For example, a security control, presented in a textual description of a system's architecture, can be a glaring copy-and-paste error because it is completely irrelevant to the system. As an example, a system has safeguards to protect Personally Identifiable Information (PII) applied to a system that does not collect, store, or transmit that type of data. Even if the security control is relevant to the system, it might be used within the wrong context. The copy-and-paste example can happen for various reasons, such as tight deadlines, lack of cybersecurity specialization, and assumed cost savings.

## 5.4 FUNDAMENTAL CONSIDERATIONS FOR MODEL REUSE

To assess the costs and benefits of a model-enabled ATO, this paper examines the potential benefits of several different objectives related to model reuse. This will partly address whether reusing an MBSE model across various similar programs creates an improved quality process and, therefore, an improved product in less time. In the context of the ATO process does this mean a more streamlined and lean procedure with improved "ilities" (reusability, portability and maintainability for example). One of the questions worth examining is how does this affect documentation? Since the data and information represented in documents do not have explicit dependencies, a change in one document needs to be reflected manually in all the other affected

documents. This manual process is not only lengthy but also prone to errors. Furthermore, when documents are used to facilitate communication, it is difficult to verify their completeness and consistency and to surface conflicting or contradictory information [65].

All models were developed using an industry-standard tool and language Cameo Systems of Systems Architect, and the Systems Modeling Language (SysML). SysML is a general-purpose system architecture modeling language for Systems Engineering applications [41].

This chapter explores how the MBSE paradigm may be leveraged to facilitate the reuse of designs and models and, more broadly, enable an organization to leverage (i.e., reuse) results from past work on future engineering projects. While facilitating reuse and achieving improved efficiency on engineering projects is a promising motivator for advancing MBSE methods, its potential has yet to be fully realized. Differences in rationales for MBSE, levels of adoption, means of deployment, and a lack of structured methodologies for exploring reuse scenarios have presented as key roadblocks [66].

### 5.4.1 THE OBJECT-ORIENTED MODEL FOR REUSE

Reuse mechanisms, such as inheritance and polymorphism in an object-oriented programming approach, are useful to professional programmers but fail to support the occasional programming needs of the end user. Consequently, a surprisingly high percentage of end users resort to "copy and paste" approaches for reuse instead of making appropriate use of object-oriented techniques. “As techniques emerge to support end-user programming and end-user modifiability, it becomes clear that end-users have little interest in programming computers unless it will help them to build tools that enhance their domain productivity [67]. Therefore, the inclusion of mechanisms to aid the end-user in the location and modification of code that performs a function similar to the one desired greatly improves the usability of domain applications. By allowing

incremental development, object-oriented languages attempt to provide reuse mechanisms such as inheritance. End-users tend to be better at thinking concretely than abstractly [67], and for this reason, inheritance works for professional programmers trained in abstraction processes but fails to work for end-users [68].” This often results in a blind copy/paste to achieve the desired effect with little understanding of how the solution works in its entirety. “Ideally in an object-oriented system, the user locates a promising class in the object class hierarchy and refines it by adding or extending methods to provide the new desired behavior. Few situations approach this ideal, however, and the result is that the user is faced with altering the object hierarchy. It is here that the process breaks down in the face of an inheritance structure defined by someone else. Some behavior of the original objects is desirable, other behavior might be merely superfluous, and still other behavior might be undesirable [68].” Thankfully, when designing systems at such a high level of abstraction, little is needed in terms of complete domain knowledge of how the underlying functions or methods in code works. In a sense, reusability becomes more feasible and potentially less complicated when modeling. However, this does not translate well to the copy/paste issues when reusing requirements.

As an alternative, this dissertation looks to reusability as an approach to enable time and cost savings by leveraging an object-oriented methodology. Object-Oriented Programming (OOP) and Object-Oriented Design (OOD) rely on four main principles: abstraction, encapsulation, inheritance, and polymorphism [69]. Reuse is a fundamental component of object-oriented programming and a major motivator for its use. Inheritance enables developers to reuse the same code multiple times by simply “calling” its function or class from another portion of the application [70]. Notice that the reuse is not a copy/paste method but a more efficient solution using good design and development practices.

Reusing software (including requirements, designs, documentation, test data and scripts, and code) improves productivity and quality, allowing us to concentrate on new problems rather than continuing to solve old ones again. A review of empirical studies of software reports that reuse led to lower problem density, decreased effort expended on existing problems, and increased productivity [71]. Regarding MBSE, the desire is to achieve the same level of rewards by leveraging the opportunities digital engineering offers. Since models are abstractions, they are digital representations of the systems and data they represent. Establishing a methodology and system to share models among projects is advantageous. Since no physical limitations exist, models are candidates for reuse due to their modality and mobility [72].

### 5.4.2 MODEL MATURATION WITH REQUIREMENTS

The model also matures as Requirements become increasingly familiar and derived requirements begin to take shape. Derived requirements result from decomposing the more significant or main requirement into smaller pieces. NIST defines a derived requirement as "a requirement that is implied or transformed from a higher-level requirement."

- Implied requirements cannot be assessed since they are not in any requirement baseline. The decomposition of requirements throughout the engineering process makes the implicit requirements explicit, allowing them to be stated and captured in appropriate baselines and allowing associated assessment criteria to be stated.
- A derived requirement must trace back to at least one higher-level requirement" [73].

Derived requirements are logical assumptions based on the more significant requirements. For example, if the system is required to operate in a "cloud architecture," a commercially approved service provider is assumed to be required. Therefore, if the cloud provider is Amazon (Amazon

Web Services or AWS), the derived requirement is that the system must operate in the AWS cloud, and the architecture must conform to that framework.

As the requirements mature and derived requirements become numerous, so will the model's structure and behaviors, which directly represent the integration of the requirements. This creates a more robust system and, regarding the ATO, a more sophisticated and mature model integrated with the latest security controls and technologies. The model can potentially become more resilient and risk-averse than the previous iterations. This is especially apparent with the application of security controls. Snapshots of the model will show the linear progression of security patches and technology integration to keep pace with the evolving threats. Modern modeling tools can display comparisons of the model at various baselines.

## 5.5 METHODS

The following paragraphs describe the approach used to develop and analyze artifacts to examine their reusability for MBSE on the ATO.

### 5.5.1 BASELINE DOCUMENTATION (TRADITIONAL) VS. MODEL TEMPLATES

Comparing the development of the same system under the baseline, or traditional document-centric approach versus model templates is not practical. There are far too many volatile variables in the ATO process to make an accurate comparison. One that presents an obvious obstacle is personnel. As stated previously, each Authorizing Official may require different deliverables and at different increments than another. Government program personnel routinely rotate on and off programs as the Government broadens its workforce. Sometimes, even development contractors are rotated out because contractual obligations were not met or to provide fair competition. However, we can make an informed comparison by examining the expected Level of Effort (LoE) to develop templates versus STE for documentation.

When looking at the LoE to develop templates, it's important to note the distinction between that and the STE when creating documentation. Developing templates is generally considered a one-time task. While templates may require periodic updates to maintain alignment with evolving standards and requirements, their fundamental structure typically remains stable after initial development and validation. Therefore, the LoE is a one-time cost associated with developing the templates. STE, on the other hand, is an ongoing cost to the program, in this example, to create the documentation and continually (manually) update it as the system matures. Each program will have STE as an incurred cost.

### 5.5.2 WHEN TO AUTO-GENERATE DOCUMENTS AND WHEN TO USE MANUAL LABOR

"Document generation is long to set up, and it requires a lot of effort to become familiar with the document generation approach and the associated tools. Investing in document generation is not an obvious solution for a project with strict deadlines and a limited budget." [61]

*"If the project has short and strict deadlines for the first deliveries, and if the question of generating documents has not been anticipated, the answer is quite simple: do not try to generate your documents from your model."* [61]

Creating a document, whether it be an architecture, interface, specification, verification plan, or any other type, will always be faster by hand the first time. This usually starts from a template or an existing document that stems from another project, then copy and paste some diagrams, extract some requirements from your requirements database, and complete it with some explanations and drawings [61]. There will be success in writing this document manually because the information to include and where to find it is known. If document generation is a new aspect of the project, there may be an uncertain amount of time required to establish everything necessary

for it to function properly. This can create considerable stress, especially if time is limited. Additionally, as noted in chapter four (4) regarding inconsistencies, incorrect or out-of-context information can lead to a range of other issues within the system. Ideally, for minor, one-off, or short-term projects, it is advisable to avoid auto-generating documents from the model simply due to the time it will take to set up and validate. Manually composing them is usually faster, as creating templates can be quite time-consuming. However, for long-term projects or those that can serve as a foundation for similar initiatives, developing templates represents a worthwhile upfront investment that can be leveraged in the future.

Chapter six (6) examines reusability with regards to a Model-Based ATO, however, it is important to note here that while the creation of custom templates requires some work, the result is reusable across all similar projects, which will benefit future projects (if the documents are similar) [61]. This is an important aspect to streamlining system deployment timelines and ensuring consistency. The approach to generating documentation from templates is illustrated below and is similar regardless of the modeling tool used.

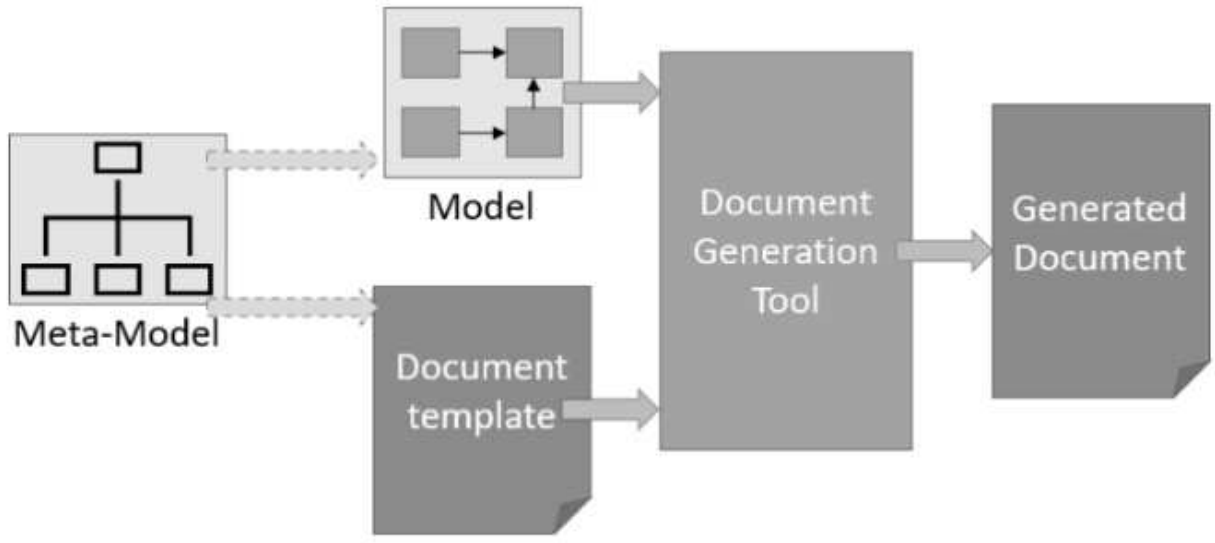



*Figure 27 - Documentation generation process from model templates [61]*

### 5.5.3 AUTO-GENERATED DOCUMENTATION

DB-ATO and MBSE-enabled ATO were compared by examining the time to develop programmatic materials. Based on our initial research into what documents are generally required for an ATO, these documents served as candidates for determining the approximate level of effort, specifically the Staff Years of Technical Effort (STE) (commonly used when costing a program). Since these programmatic documents are often closely related and rely on information provided and referenced to each other to describe a complete system, it is noticeable how an inconsistency in one can affect another. Therefore, when initially costing a system, it does not account for the number of man-hours needed to verify and validate the requirements through traceability to test cases and inconsistent procedures. The inconsistency will lead to confusion and lost time to interpretation and corrective action. This, in turn, can affect the schedule, which impacts cost. Often, staying on schedule due to varying pressures will lead to cutting corners, which only exacerbates the problem. Referring to Table 3, 41 documents are identified as candidates for template generation and document reuse, where inconsistencies can permeate throughout if not identified and appropriately managed.

Recall that the documentation highlighted in blue has been identified as candidates for creating templates for auto-generating documents. The candidacy for template development is due to the frequency with which these documents are used in all USG ISs. When a template is used, time is saved, and attention can be spent on the detailed engineering of the system. Templates also reduce the potential for inconsistencies since the documents are pre-formatted and provide a layout that specifies which type of content is expected where. There is less manual inputting of information and data, which reduces the margin of human errors in both the creation process and overall document formatting. Also, with an automated system, there is less need for repetitive

cutting, pasting, and double-checking for errors in documents [54]. This will also create uniformity of the documents. These documents can also be autogenerated from any other model tool that supports the functionality. The documentation highlighted in red has been identified as candidates for architectural reuse due to the USG cloud-provided development tools. The last column represents the total number of documents associated with a system. However, that number can fluctuate greatly depending on the system's complexity and other factors.

Examining Table 3 shows that specific documents are associated with explicit process steps in the ATO/RMF. The column “ATO Documentation Package – Required” shows that there are nine (9) documents that must be delivered, per system, for accreditation. These documents are the: System Definition Document, System Security Plan, Updated System Security Plan, Status Report, Security Assessment Report, Security Assessment Plan, POA&M, Risk Assessment, and Monitor Strategy Document. Because those are required documents, they are considered candidates for templating and should be auto-generated whenever possible. Due to the similar characteristics the Asset Management System shares with other Government, cloud-centric systems, it is used to demonstrate the savings of documentation reuse. Furthermore, as the system progresses through the ATO process, we can track its progress and anticipate the next deliverables or milestones.

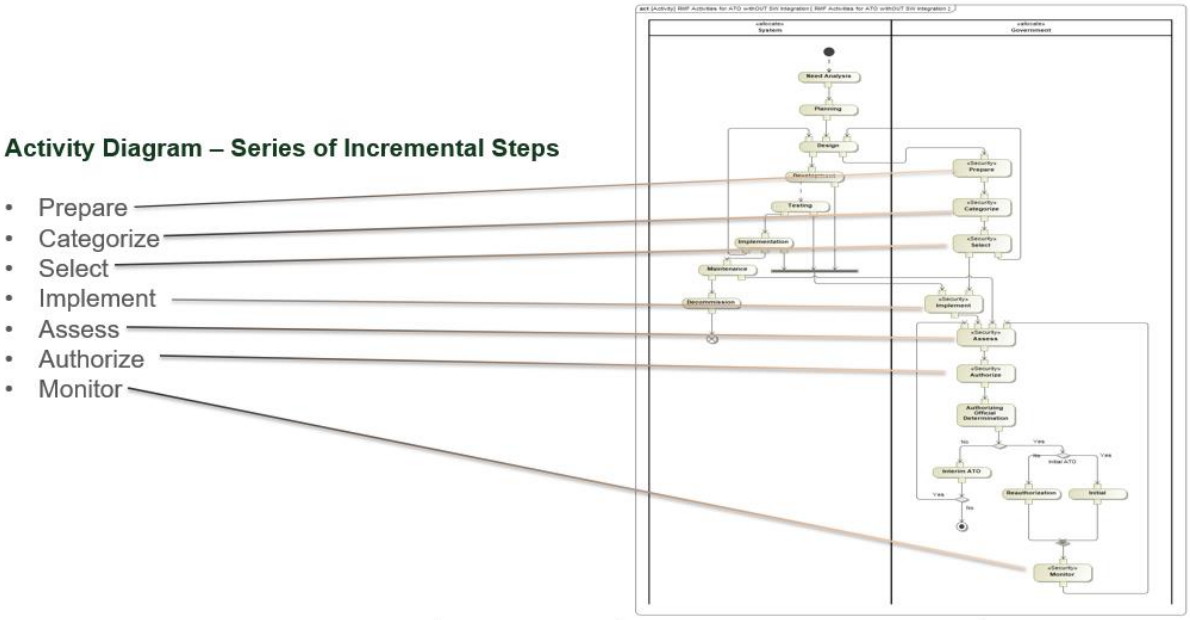



*Figure 28 - Activity Diagram with ATO/RMF Process Steps*

Examining the documentation that aligns with the ATO/RMF step, it helps to explain why, whenever possible, it is best to reuse documents that have previously proven effective and to improve consistency and reduce time to accreditation.

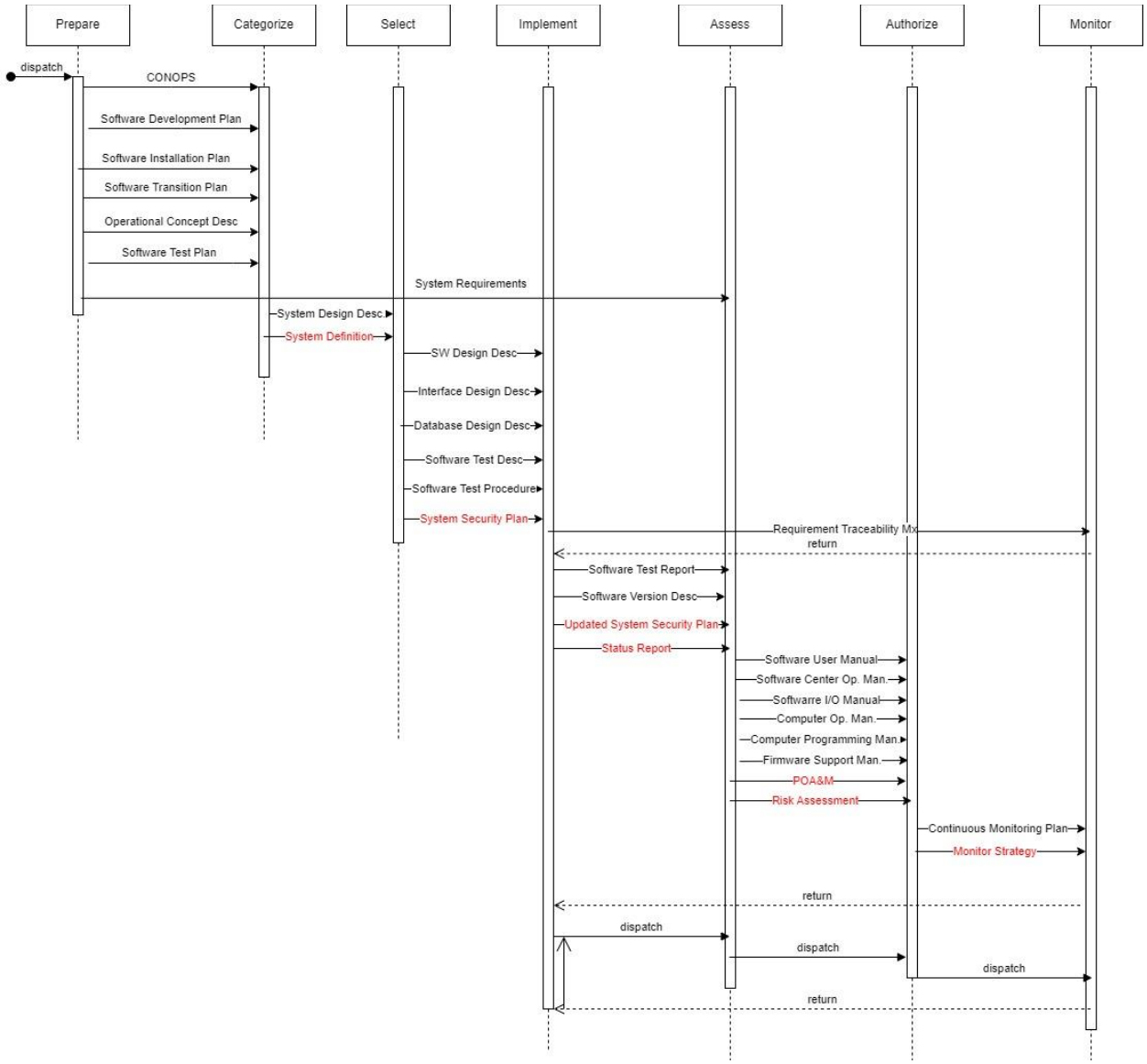



*Figure 29 - ATO Documentation by Process Steps of the RMF*

In Figure 29 - ATO Documentation by Process Steps of the RMF above, the documents in red font are required by the RMF for accreditation. Because these documents are required, they should be generated from templates to improve consistency, reduce redundancy, increase efficiency, and potentially reduce costs. It is important to note that the diagram does not represent iterations of the documentation. Therefore, time to create, review, edit, and peer review are not displayed. The iterations of the documentation (i.e., different versions such as “draft” to final product) require time, and it’s not uncommon for that time to be considerable.

### 5.5.4 TYPES OF DOCUMENTS THAT CAN BE AUTO-GENERATED

The primary documents generated from a system model concern the system architecture: list of functions, functional breakdown, functional architecture, components, product breakdown structure, interfaces, functional behavior, and automation. Often, the description of needs and expected functionalities: use cases, context diagrams, external interfaces, scenarios, some state machines, and the generation of ICDs, either as text (Microsoft© Word documents) or as Microsoft© Excel spreadsheets [61].

*"With tools such as Cameo, this type of information that is gathered from the model can be exported as a document. In Cameo Systems Modeler, the document generation tool is called Report Wizard. It is a technology developed specifically for Cameo Systems Modeler and comes natively with this tool. Report Wizard is based on Velocity, a Java-based template engine. It requires templates to be written in Velocity Template Language (VTL)."*

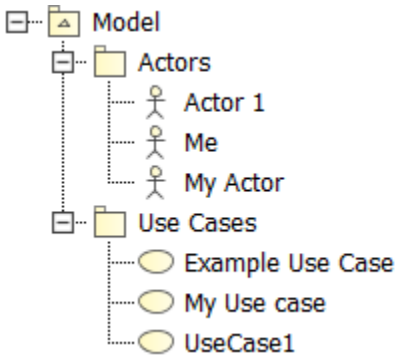
To generate a document from a model using the *Report Wizard*, it is enough to open the *Report Wizard* from the model that the document should be generated from and selecting the desired template. Just like any other wizard, it guides the user through the steps of the documentation creation. *Cameo Systems Modeler* comes with a wide selection of templates, though in order to obtain a document that is truly useful, custom templates will be necessary. However, these existing templates do give good starting points for developing custom templates.

The *Velocity Template Language* is written in plain text directly in the template document, where the formatting applied to the template queries (code) reflects the formatting in the finalized generation.

In *VTL*, each variable is prefaced with "$", and each command line to be executed starts with "#". Any lines of text not prefaced with "#" will be reproduced in the generated document, though any variables (starting with "$") will be replaced with their value.

In addition to the basic queries and operations native to *VTL*, some helper modules and special variables have been developed specifically for use with *Cameo Systems Modeler*, that make some of the information in the model much easier to access. For instance, the variable *$elements* is a list of every single element that exists in the scope of the model selected for generation, and the helper module *$report* makes it possible to obtain a filtered list of elements.

For example, if the model looks like this:




Using this template:

```
All Use Cases and Actors
#foreach($element in $report.filterElement($elements, ["UseCase", "Actor"]))
    • $element.name: $element.humanType
#end
```

Will give this result:

All Use Cases and Actors

- **Example Use Case**: UseCase
- **My Use case**: UseCase
- **Actor 1**: Actor
- **My Actor**: Actor
- **Me**: Actor
- **UseCase1**: UseCase

*VTL* can be used in many different formats, including Microsoft© Word, Excel, PowerPoint and Hyper Text Markup Language (HTML). In this dissertation, the focus is on the generation of Microsoft© Word documents" [61].

To illustrate why using templated documents for similar projects is advantageous, the Sequence diagram below shows the steps taken between the document author and the respective Government customer. The process is repeated for each document created, as shown in Figure 30 - Documentation Sequence Diagram. Eliminating even a few documents can save many hours.

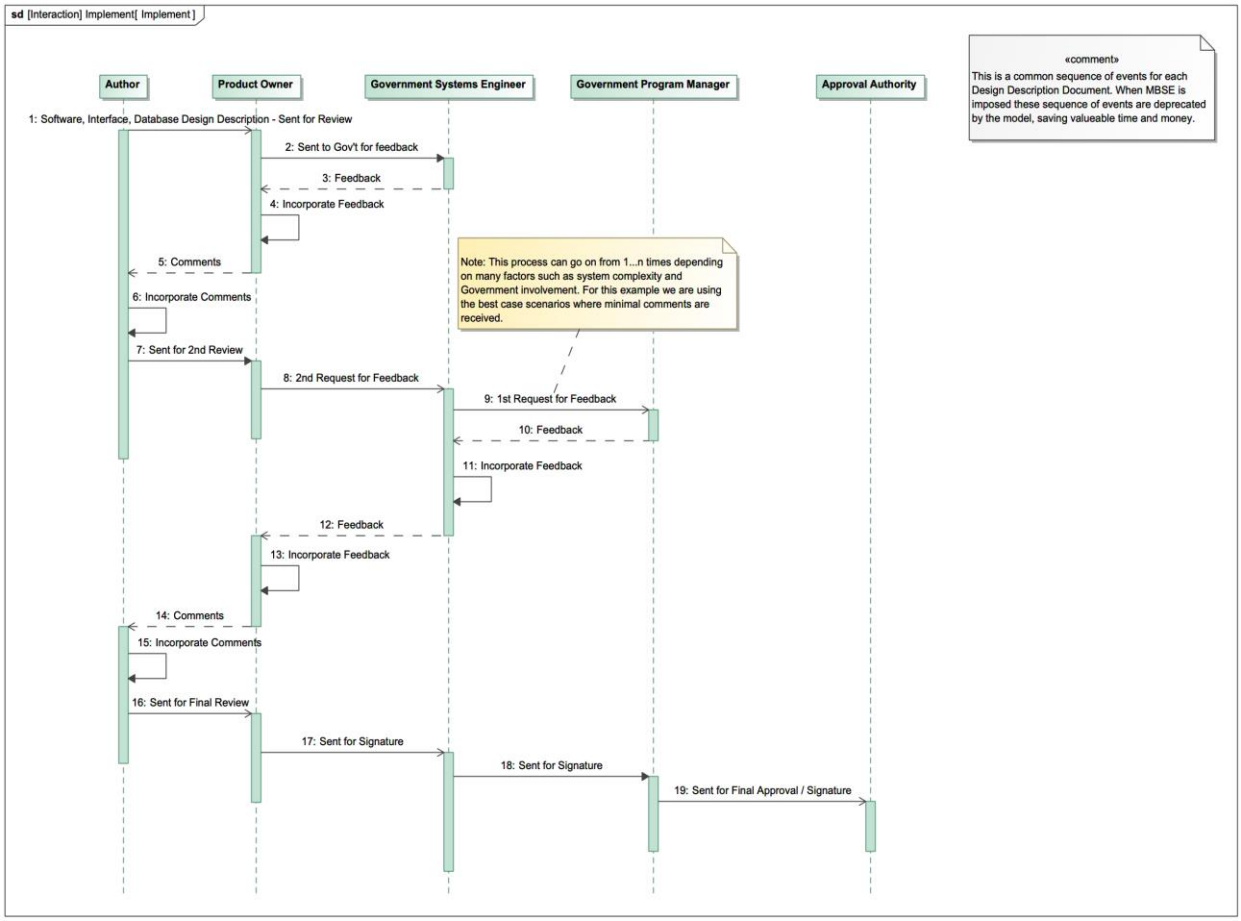



*Figure 30 - Documentation Sequence Diagram*

This model of a USG IS is based on the specifications and guidance from the DoD Cloud Strategy [44]. This fictitious system uses the resources available as approved by the USG for cloud application development. These tools are commercially available and have undergone a thorough

security analysis before being adopted and integrated into the USG's cloud infrastructure. The accreditation process for the commercially available tools, even from trusted vendors, must also follow the ATO process. Therefore, there is often a lag between developing commercially available tools and their accreditation and availability in the USG cloud. As such, this IS (a fictitious Asset Management System) accurately represents the architecture and functions found in current cloud computing USG systems. The system is presented as a Block Definition Diagram (BDD) in Figure 6. Critical components of this structure/data view of the IS include key personnel, hardware, software, and documentation. Generally, a System Element Specification [32] can accompany the BDD to provide additional information specific to each block.

An example is given for the Project Staff Block of the IS in Table 9 - System Element Specification - Project Staff. Within these models are operational/functional and requirements models for the IS. The model in its entirety is available for distribution at www.engr.colostate.edu/se

Using this model of the IS, we can represent the process by which similar cloud-based systems will proceed through the ATO. We will be able to measure the effect of architecture reuse, the capabilities to reduce inconsistencies, and an opportunity to manipulate the impacts of security controls.

### 5.5.5 DOCUMENTS GENERATED FROM THE MODEL TEMPLATES

For the foreseeable future, system documentation will continue to exist. The primary documents generated from a system model concern the system architecture: list of functions, functional breakdown, functional architecture, components, Product Breakdown Structure, interfaces, functional behavior, and automatons. Eight (8) documents are required as part of the ATO. Those documents are explicitly listed in Table 1 in the column "ATO Documentation

Package – Required." Those documents are the System Definition Document, System Security Plan, Updated System Security Plan, Status Report, Security Assessment Report, Security Assessment Plan, Plan of Action and Milestones (POA&M), Risk Assessment, and Monitor Strategy Document.

These documents are excellent candidates for reuse, particularly from templates, as they are essential for any system seeking accreditation. A natural consistency emerges when documents are generated from the system model. Configuration management is improved as templates are created and stored in a repository offered by many commercially available modeling tools. For the purposes of this paper, the models and templates were developed using a combination of Cameo's System Modeler and Systems of Systems Architecture.

Validating the reports, which directly reflect the model, is expected to take minimal time. The model is designed to ensure the accuracy of the report content before it is generated from the template.

## 5.6 RESULTS

The following paragraphs discuss the reusability results in the ATO and MBSE context. They provide examples of implementations and examples of element portability in the form of stereotypes, block elements, and NIST SP 800-53 Rev. 5 security controls. Lastly, this section describes model-based requirements, architectures, design patterns, and continuous security for reuse.

### 5.6.1 EXAMPLE IMPLEMENTATIONS

One of the most frequently highlighted advantages of reuse is the significant potential for saving time and money by minimizing, or ideally eliminating, the duplication of efforts. In the realm of software systems, code, and libraries have been reused across different applications for a

long time, enhancing consistency and standardization among those applications. Similarly, application frameworks and design patterns have seen extensive reuse.

When MBSE is applied effectively, all stakeholders can reference a unified model that captures the system's context and architecture. Instead of each stakeholder creating entirely new use cases, block diagrams, or documentation, they can contribute to updating the architect's model to reflect their concerns and document the outcomes of their analyses. Configuration Management can become cumbersome as the number of documents and their respective versions increases. Consequently, the opportunity for cost savings grows with each additional stakeholder who adopts the architecture model as their primary reference.

#### 5.6.1.1 ELEMENT PORTABILITY

When developing unique elements for system modeling that aren't readily available in the tool, it's crucial to prioritize reuse. Tools like Cameo facilitate the creation of custom elements that can be utilized throughout the model and in other system models for different projects. For instance, the Asset Management System includes ten customized stereotypes that can easily be applied to other models. Given that these stereotypes share common characteristics, it's highly likely that they can be employed repeatedly, eliminating the need for recreating them each time.

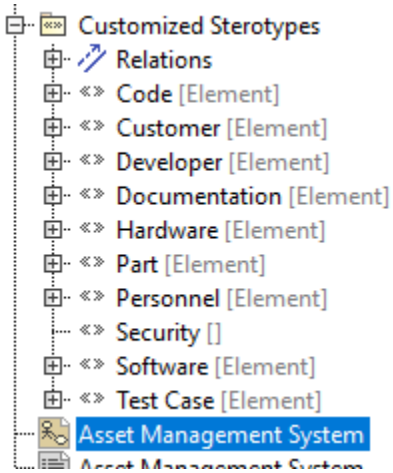



*Figure 31 - Custom Stereotypes*

When appropriately designed, and utilizing the components of the Asset Management System for an evaluation, it is estimated that nearly 90% of the model can be reused across similar systems. This high level of reusability is facilitated by the cloud-centric mandate, supported by the Federal Risk and Authorization Management Program (FedRAMP) and Cloud Smart initiatives. The only elements that account for the estimated 10% non-reusability are unique system artifacts. For instance, the Asset Management System developed for this research includes BDDs that are applicable for use in comparable systems.

#### 5.6.1.2 SECURITY CONTROLS (NIST SP 800-53 REV. 5)

The NIST Special Publication 800-53 Revision 5 outlines all the security controls that have been identified to secure systems alongside the RMF. The NIST Special Publication 800-53, *Security and Privacy Controls for Information Systems and Organizations,* describes the security controls in detail. These are broken down into control families, of which there are 20 [74]. Figure 32 - NIST Security Control Families lists the NIST Security Controls by Identification (ID) and family name.

| ID | FAMILY | ID | FAMILY |
|---|---|---|---|
| AC | Access Control | PE | Physical and Environmental Protection |
| AT | Awareness and Training | PL | Planning |
| AU | Audit and Accountability | PM | Program Management |
| CA | Assessment, Authorization, and Monitoring | PS | Personnel Security |
| CM | Configuration Management | PT | PII Processing and Transparency |
| CP | Contingency Planning | RA | Risk Assessment |
| IA | Identification and Authentication | SA | System and Services Acquisition |
| IR | Incident Response | SC | System and Communications Protection |
| MA | Maintenance | SI | System and Information Integrity |
| MP | Media Protection | SR | Supply Chain Risk Management |

*Figure 32 - NIST Security Control Families*

Below is an example of their organization. This example is an excerpt from the System and Information Integrity section.

| Control Number | Family | Control Title | Control Text | Confidentiality | Integrity | Availability | Supplemental Guidance |
|---|---|---|---|---|---|---|---|
| | | TOOLS | | | | | |
| SI-4 (10) | SI | INFORMATION SYSTEM MONITORING \| VISIBILITY OF ENCRYPTED COMMUNICATIONS | The organization makes provisions so that [Assignment: organization-defined encrypted communications traffic] is visible to [Assignment: organization-defined information system monitoring tools]. | High Moderate | High Moderate | High Moderate | Supplemental Guidance: Organizations balance the potentially conflicting needs for encrypting communications traffic and for having insight into such traffic from a monitoring perspective. For some organizations, the need to ensure the confidentiality of communications traffic is paramount; for others, mission-assurance is of greater concern. Organizations determine whether the visibility requirement applies to internal encrypted traffic, encrypted traffic intended for external destinations, or a subset of the traffic types. |
| SI-4 (11) | SI | INFORMATION SYSTEM MONITORING \| ANALYZE COMMUNICATIONS TRAFFIC ANOMALIES | The organization analyzes outbound communications traffic at the external boundary of the information system and selected [Assignment: organization-defined interior points within the system (e.g., subnetworks, subsystems)] to discover anomalies. | High Moderate Low | High Moderate Low | High Moderate Low | Supplemental Guidance: Anomalies within organizational information systems include, for example, large file transfers, long-time persistent connections, unusual protocols and ports in use, and attempted communications with suspected malicious external addresses. |
| SI-4 (12) | SI | INFORMATION SYSTEM MONITORING \| AUTOMATED ALERTS | The organization employs automated mechanisms to alert security personnel of the following inappropriate or unusual activities with security implications: [Assignment: organization-defined activities that trigger alerts]. | High Moderate Low | High Moderate Low | High Moderate Low | Supplemental Guidance: This control enhancement focuses on the security alerts generated by organizations and transmitted using automated means. In contrast to the alerts generated by information systems in SI-4 (5), which tend to focus on information sources internal to the systems (e.g., audit records), the sources of information for this enhancement can include other entities as well (e.g., suspicious activity reports, reports on potential insider threats). Related controls: AC-18, IA-3. |
| SI-4 (13) | SI | INFORMATION SYSTEM MONITORING \| ANALYZE TRAFFIC / EVENT PATTERNS | The organization:<br>(a) Analyzes communications traffic/event patterns for the information system;<br>(b) Develops profiles representing common traffic patterns and/or events; and<br>(c) Uses the traffic/event profiles in tuning system-monitoring devices to reduce the number of false positives and the number of false negatives. | | | | |

*Figure 33 - Example NIST Security Controls*

There are over one thousand NIST security controls available for consideration. For this research, these controls were initially provided in Microsoft Excel format and subsequently imported into Cameo using the Excel/CSV import plugin. Once imported as a distinct project in Cameo, the security controls were merged into the Asset Management Information System (IS). With the security controls established as their own project, they can now be reused across an infinite number of system development initiatives. This enhances their portability, manageability, and reusability.

Moreover, the integration of these security controls requires a thorough examination, as each system may necessitate the incorporation, assessment, and testing of hundreds of controls to achieve an acceptable level of security for accreditation purposes. There are three established security control baselines corresponding to each system impact level—low-impact, moderate-impact, and high-impact—as well as a privacy baseline applicable to systems regardless of their impact level. Of these controls, 370 are classified as "high" in terms of their importance for securing systems. Therefore, reusing these security controls as model elements is considered highly advantageous.

Ideally, each security control will be modeled and usable as an enterprise solution when designing systems. For this paper, a subset was taken from the example in Figure 33 to show what that may look like. Once the security controls are modularized, which is the case for this dissertation, they can be reused 100% of the time. This creates the capability to "drag and drop" into the system models or show allocation and traceability to them in the matrixed form as show in Figure 34. Maintainability should be minimal and only occur if / when the specific control is modified. Therefore, it is plausible to say that regarding securing Government ISs, the portions of the model required for its security would not have to incur the cost to develop. This saves a considerable amount of time and money which means the potential for quicker development of operational capabilities. Below is an example of how the modularized security controls can be integrated as part of the Government IS.

As previously noted, the reuse of effective security controls can significantly enhance the security posture of a system and potentially expedite delivery to operations, such as production or forward-facing environments. For this research, a comprehensive spreadsheet of all NIST SP 800-53 Rev. 5 security controls (which includes over one thousand items) was integrated into the Magic Systems of Systems Architect 2022x application. With this integration now complete, the package containing all 1000+ security controls are available for sharing across the enterprise. Subsequently, these security controls were mapped to the system element blocks that require protection. Figure 30 illustrates an example of this mapping.

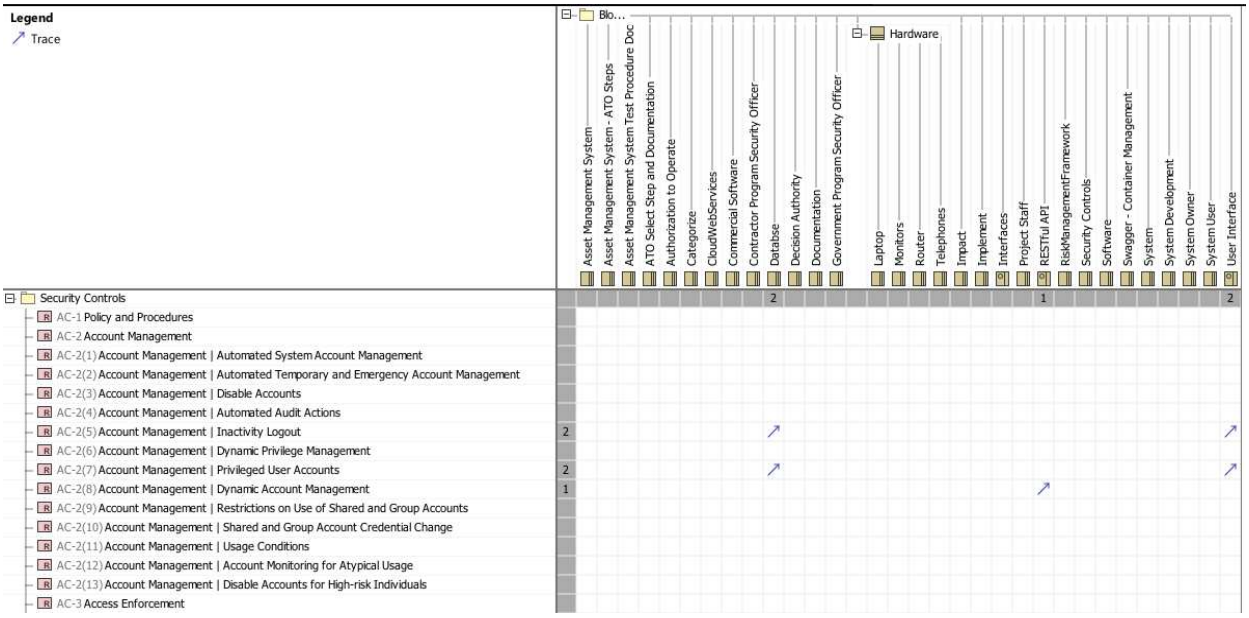



*Figure 34 - NIST SP 800-53 Rev.5 Security Controls Allocated to the Asset Management System Block Elements*

The left side of the display features the security controls imported from the NIST SP800-53 Revision 5 spreadsheet. The elements at the top represent block components of the example Asset Management system. A blue arrow indicates when a specific security control is necessary to safeguard a portion of the system. The figure shows that the security control AC-2(5) Account Management | Inactivity Logout is implemented on both the Database and User Interface as a security measure.

#### 5.6.1.3 MODEL-BASED REQUIREMENTS FOR REUSE

"Often, requirements are considered artifacts of a project useful only for the project's duration. Once the change is executed, the system is modified, or a new process is established, data and process requirements documentation is filed away in an archive folder and forgotten. Not only is this valuable to the organization's knowledge management, but implemented requirements can be reused in subsequent initiatives to:

- Describe the current state of a business process, function, or capability,

- Define relationships between business concepts and its data management,
- Reflect the needs of internal and external personas,
- Capture key business rules that must either be maintained by the new solution or consciously changed,
- Become templates or starting points for new requirements.

Requirement reuse can significantly reduce analysis time and improve the overall quality of business analysis outcomes [75]."

Similar systems can often expect similar requirements, especially when the system is required to operate in a cloud-centric architecture, which is required for US Government ISs with only a few exceptions. When Requirements are modeled, their references are checked for accuracy and consistency, and reference material can be inserted as part of the model. This is best demonstrated back in Figure 20.

Here, it can be observed that not only the structure of the tests and how the requirements are satisfied but also a direct link to the test documentation embedded in the *Package* element in Cameo. For projects that are hesitant to transition to a fully MBSE development process, Figure 24 demonstrates how even greater traceability can be achieved even when tests are developed in a document-centric style. Furthermore, those documents can also be created from templates from the model, as discussed previously.

#### 5.6.1.4 ARCHITECTURES AND DESIGN PATTERN REUSE

Because of the mandate for the Government ISs to leverage cloud computing and technology, there is an increase in similar application design patterns. For the most part, this should be expected as the tools used to design, build, and run the IS all result from the integration of pre-

approved Commercial of the Shelf (COTS) and Government of the Shelf (GOTS) software and hardware.

Referring to the Asset Management System Block Definition Diagram (BDD) below, and referring back to 5.6.1.1, it is realistic to estimate that 90% can be reused due to its cloud architecture, which is common among US Government ISs. The other non-reusable 10% are system-specific elements or characteristics. Creating a generic cloud-centric model for reuse saves considerable time and serves as a backbone for designing a multitude of systems.

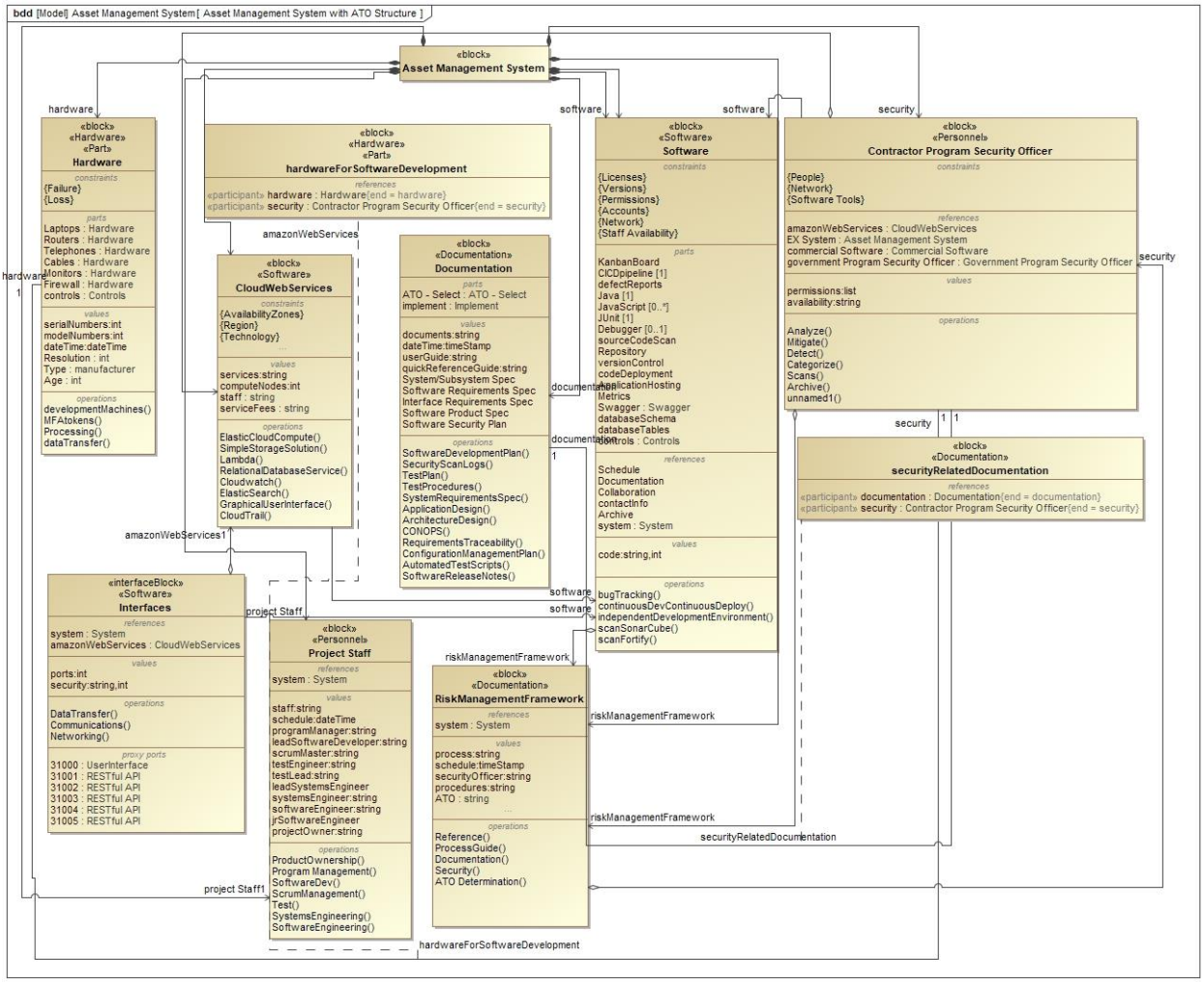



*Figure 35 - USG IS Model in the form of a SysML Block Definition Diagram*

"Some tools, like *Cameo Systems Modeler*, have the capability to compare two versions of a model and to generate the differences between them in a document. In this way, document generation can

be a tool for analyzing the evolution of the models between 2 versions [61].” This is an effective way to see how the model matures over time and how new or changing requirements influence the system and its structure/behavior. This analysis, when showing the changes in the model itself, can help identify potentially new areas of concern regarding security and risk management.

#### 5.6.1.5 MBSE DESIGN DECISIONS

According to the comprehensive analysis presented in the Sandia National Laboratories Report SAND2016-2607 “Systematic Literature Review: How is Model-Based Systems Engineering Justified,” Chodas justified an MBSE approach by illustrating improvements in cost and schedule in his case study, “Improving the Design Process of the REgolith Imaging X-ray Spectrometer (REXIS) Using MBSE.” He set the stage by showing (Figure 36) that NASA had a significant number of projects with cost and schedule overruns [76].

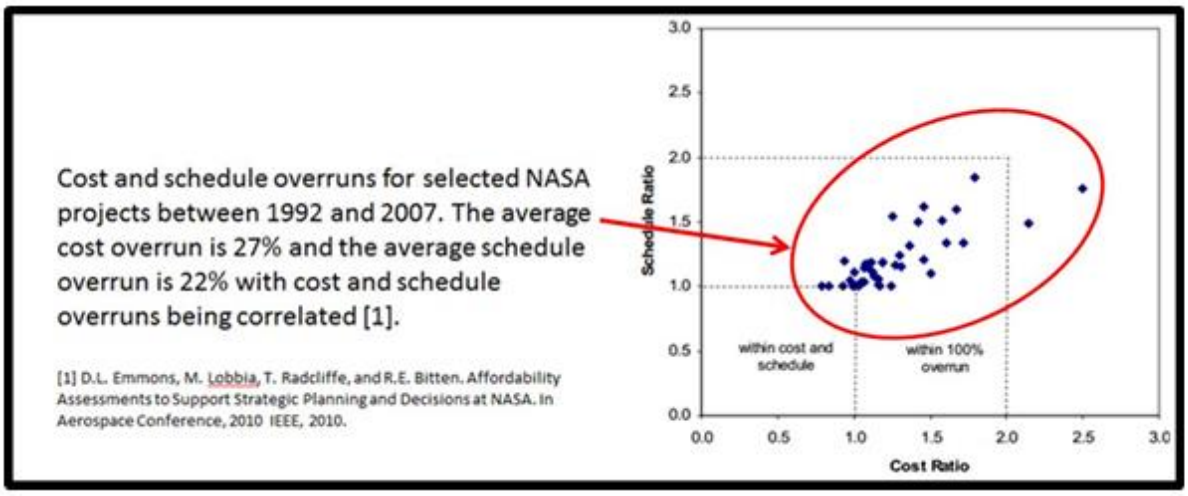



*Figure 36 - Cost Overruns at NASA*

Chodas explained that many systems at NASA are system-of-system projects and attributed the cost and schedule overruns to the growth of new subcomponents (parts) added to the system design in order to solve problems found as a result of design defects, omissions, or changes. He showed that an MBSE approach can have a significant positive impact on project cost and schedule by limiting the amount of rework as an improvement above using a traditional DBSE approach.

Figure 37 illustrates how component (parts) growth often occurs after System Design Review (SDR), a point after which it becomes increasingly more costly to make changes [24].

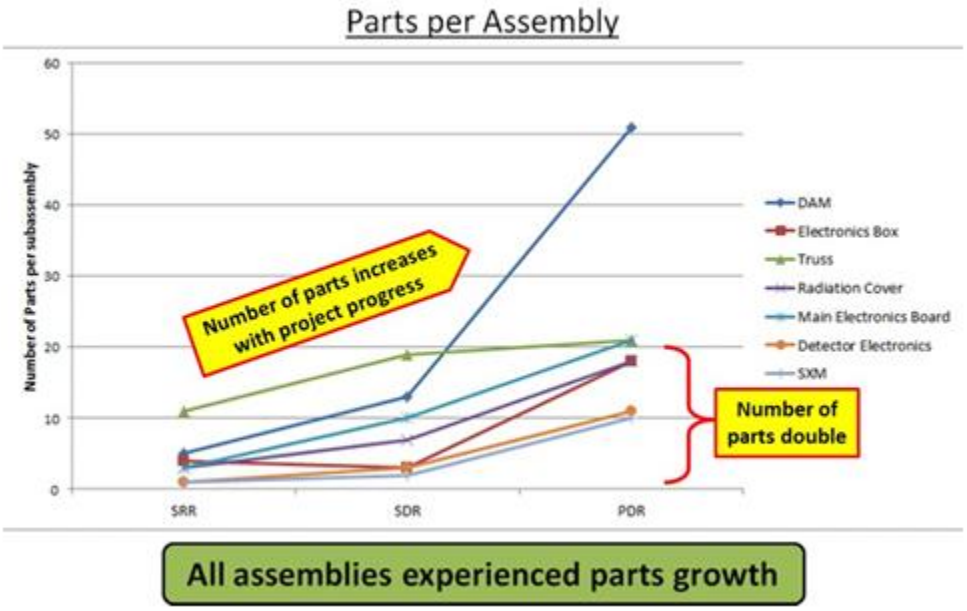



*Figure 37 - Part Growth After System Design Review*

In Figure 37, Chodas provides an example of how his team successfully used an MBSE approach to find a design solution sooner than they would have found using a DBSE approach [24].

#### 5.6.1.6 REUSE FOR CONTINUOUS SECURITY

One of the RMF's steps is “Assess.” This step, however, is not completed once. Instead, because the ATO requires reaccreditation every few years, assessing the security controls and their effectiveness is ongoing. The Assess step is represented in Figure 38 as part of the Continuous Monitoring package.

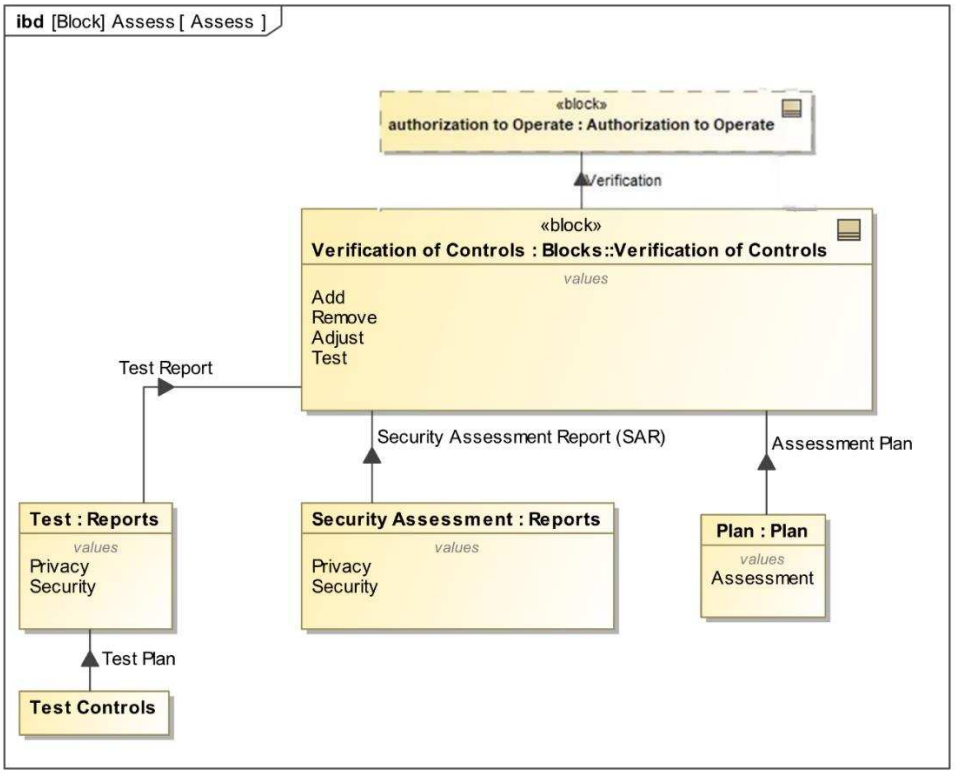



*Figure 38 - Asset Step Internal Block Definition Diagram*

To ensure continuity of information, it is incumbent upon agencies to thoroughly assess their operational, policy, and business requirements and advocate for themselves when brokering new arrangements with cloud service providers and their internal security processes. Now that the 2019 Federal Cloud Computing Strategy, or Cloud Smart is in place, adopting and implementing a zero-trust approach to cloud vendors is recommended. Though FedRAMP has reduced the amount of time it takes to authorize a cloud service prover, the process for assessments continues at a slow pace.

*"FedRAMP provides a standardized government-wide approach to security assessment, authorization, and continuous monitoring of cloud services. Offering cloud service providers, the opportunity to demonstrate their ability to meet Federal security requirements through standardized baselines has allowed for a flourishing marketplace of vetted providers to develop. It has also allowed agencies to adapt from arcane legacy technology to mission-centric and cost-effective cloud-based systems in a more rapid, consistent, and secure manner."* [79]

Perhaps the greatest take away is the statement below taken from the Federal Cloud Computing Strategy, and precisely where MBSE is expected to mitigate these challenges.

*"…a lack of reciprocity across agencies when adopting FedRAMP authorizations has led to significant duplication of effort when assessing security for product deployment. In addition**, a large number of agency-specific processes have made it complicated for agencies to issue an Authorization to Operate (ATO) for solutions**, **even when using existing authorized cloud service providers.** In fact, despite the reiterated importance of enterprise risk management, agencies continue to cite major obstacles with their own policies and practices."* [77]

Aside from the security controls offered by the cloud provider, reusing custom or Commercial off-the-shelf (COTS) could reduce the time to reassess and offer additional protections to similar systems gathered from their successes. Regardless of provider type – commercial or Federal – agencies should consider having agreements with all providers regarding access to and use of log data, given its importance in effectively conducting information security operations. The logs provide insights into how the system managed threats, the number of threats over time, and other valuable data. If the security control is proven successful, it should be reused. Moreover, as each agency is the custodian of its information on behalf of the public, each agency is responsible for determining its governance model for cloud-hosted data that aligns with its identity and credential management systems. To that end, when a vendor deploys a cloud solution, a Service Level Agreement (SLA) should be in place that provides the agency with continuous awareness of its information's confidentiality, integrity, and availability [77].

*"Critical to the success of this security strategy in the context of Cloud Smart is the assurance of confidentiality, integrity, and availability of Federal information as it traverses networks and rests within systems, regardless of whether those environments are managed locally, off-premises, by a government entity, or by a contractor."* [77]

## 5.7 DISCUSSION

The following subparagraphs provide a summation of the results of leveraging MBSE for the purposes of reusability.

There is little doubt that reusability is essential in system modeling and development. However, quantifying the associated savings can be challenging due to the multitude of variables

involved, including social and interpersonal factors that are often difficult to measure. Each variable influences the time and cost required to deliver systems to an operational environment. What is clear is that a system that has been granted an ATO is an excellent candidate for reuse, where applicable. MBSE enhances the potential for successful reusability through data centralization and the application of object-oriented best practices. The quality of the product, particularly regarding security, is improved as the necessary controls have already been tested and an ATO has been issued. Additionally, the time to deploy to the operational environment is shortened; however, due to the specifics of each system, it is challenging to quantify this reduction. Documentation also becomes more accurate, as it is generated directly from the system model through various templates and visual transformation languages (VTL). These templates are populated with data sourced directly from the model.

### 5.7.1 CHALLENGES WITH COMPARISON

As each system is unique, so is each process for accreditation through the ATO. Personnel, technologies, schedules, funding, and a myriad of other factors play into reasons while a direct comparison of an MBSE approach to the baseline document-centric ATO is not entirely possible. It is best to take a quantifiable and realistic estimation based on known variables. It is through this process that we can measure the potential benefits of an MBSE approach to accreditation.

Correct reusability is a challenge, but when done properly can yield rewards with consideration to cost, schedule, and maintainability. Adopting MBSE and placing the model and data at the core of system development is a good start. Using MBSE to assist with the ATO and to help identify and mitigate the many pitfalls that can create an undesirable outcome later is encouraging. MBSE will encourage open dialog among all involved in the system's lifecycle.

## 5.8 CONCLUSION

Although the initial investment in adopting MBSE can be substantial, the cost savings gained by minimizing duplicate efforts among various stakeholders during the development of complex systems can outweigh this investment when MBSE is implemented effectively. By utilizing proper reuse strategies and adopting documentation templates, the number of redundant tasks can be significantly diminished. This leads to a more streamlined and efficient development process, potentially resulting in faster deployments and enhanced system maturation.

# CHAPTER 6 – MODELING FOR QUICKER DEPLOYMENTS AND COST SAVINGS

## 6.1 DESCRIPTION OF RESEARCH QUESTION THREE (3)

MBSE has been recognized for its ability to reduce the schedule (specifically the time to develop systems) and costs associated with systems engineering development programs. This cost reduction can be attributed to several factors, including minimizing the risk of errors and inconsistencies, improving traceability, and enabling reuse at multiple levels. The results and conclusions derived from RQ1 and RQ2, have provided evidence that MBSE processes, models and tools can be tailored to enable a MBSE-based ATO. This final chapter of this dissertation seeks to understand whether this model-based ATO process can realize the improvements attributed to MBSE, specifically whether a model-based ATO can reduce the time to achieve accreditation and deployment and whether it can thereby reduce the cost of accredited USG ITSs.

Research Question Three (3) - *Can MBSE reduce the overall cost of an ATO program by enabling quicker releases? Does the reuse of models not only promote more timely releases but does the accuracy and predictability of the security controls for similar threat scenarios based on artifacts lead to less overhead?*

This chapter tackles the research question by outlining a series of research tasks. These tasks will facilitate a discussion on the impacts of MBSE on the ATO and whether the findings support a reduction in costs due to faster deliveries compared to the traditional DB-ATO process, which has prevailed in the industry for over half a century.

A pertinent question arises: Does the reuse of a system model accelerate the delivery of operational solutions? Additionally, are costs reduced as a result of a more streamlined process?

Many software systems in the federal sector are now functioning in secure cloud environments using common services and components. Will the reuse of IS models that share a common cloud architecture lead to faster deployments, given the existing model and the security controls that have already been identified? For instance, if System A is a cloud-based software solution that has been built, granted an ATO, and deployed operationally using a specific model, can System B benefit from reusing that model as a foundation for its own development and authorization? This approach could potentially reduce costs as a byproduct of utilizing a previously accredited model.

## 6.2 INTRODUCTION

Quantifying the savings in time and resources from applying an MBSE methodology and associated process refinements can be tricky because they vary from project to project. However, evidence of such payoffs is emerging in various system categories. In a recent case study, two software development teams worked on the same project in parallel. One used the DBSE method, and the other used Agile / MBSE. Both teams tracked the time it took to complete the solution. The result was a 71% reduction in labor hours using MBSE compared to the document-based approach [78].

Another study compared using an agile model-based software engineering approach versus simply agile, specifically Scrum, and observed "commitment reliability and productivity for Scrum Model-Based System Architecture Process (sMBSAP)-driven sprints were larger than those of the scrum-driven sprints, and the observed defect rate for the sMBSAP-driven sprints was smaller than that of the scrum-driven sprints [79]."

"Specifically, it was observed that there was a 16% increase in the CR, a 13.4% increase in the SV, a 22.5% increase in the CLOC per hour, a 31.8% decrease in the DD using the PBIs method, a 50% decrease in the DD using the KLOC method, and a 21.4% decrease in the DL."

- Commitment Reliability – 16% increase
- Sprint Velocity – 13.4% increase
- Count of Lines of Code – 22.5% increase per hour
- Defect Density – 31.8% decrease using the product backlog items method and 50% decrease using the kilos (thousands) of lines of code method

The improved reliability of estimation, productivity, and defect rate could potentially help reduce the risk of running behind schedule and overbudgeting that can occur with agile-driven projects. Overall, these results provide some evidence of the efficacy of a combined agile MBSE approach in managing software-based systems and in strengthening the case for its adoption within the software development community, as well as the broader systems engineering community [79].” For the purposes of this dissertation, it is expected this can be directly translatable to MB-ATO supported by examples presented thus far and following.

There remains little information for costing analogies for MBSE on large programs because of the proprietary nature of project costing data [80] [81]. There have been some studies that show that MBSE will significantly enhance cost savings, particularly when control assessments are performed in accordance with the assessment plan. This information can be obtained by reviewing the model, eliminating the need to sift through extensive program documents. The model offers a clearer understanding of where controls are most effective by presenting a realistic visualization of the system rather than relying solely on documentation interpretation. A reasonable estimate suggests that the time required for control assessments could be reduced by approximately 10% to 15%, based on the quantifiable data outlined in the table below [82].

*Table 4 - MBSE Cost Savings [82]*

| Worldwide | 2015 | | 2013 | | 2010 | |
|---|---|---|---|---|---|---|
| **Using MBSE** | ✓ | × | ✓ | × | ✓ | × |
| Development time months | 13.2 | 12.7 | 8.5 | 13.4 | 12.9 | 11.7 |
| % Behind schedule | 31.5% | 38.6% | 38.7% | 38.8% | 45.6% | 56.5% |
| Months behind | 5.2 | 4.9 | 5.9 | 4.9 | 4.2 | 3.9 |
| % Cancelled | 8.9% | 16.3% | 11.1% | 12.7% | 11.4% | 14.3% |
| Months lost to cancellation | 4.1 | 4.3 | 6.0 | 5.4 | 5.4 | 4.3 |
| SW developers/project | 4.8 | 10.4 | 8.5 | 13.4 | 8.9 | 12.4 |
| Average developer months/project | 63.4 | 132.1 | 72.3 | 179.6 | 114.8 | 145.1 |
| Dev. months lost to schedule | 7.9 | 19.7 | 19.4 | 25.5 | 17.0 | 27.3 |
| Dev. months lost to cancellation | 1.8 | 7.3 | 5.7 | 9.2 | 5.5 | 7.6 |
| Total developer months/project | 73.0 | 159.0 | 97.3 | 214.2 | 137.3 | 180.0 |
| **At $10,000/developer month** | | | | | | |
| Average developer cost/project | $633,600 | $1,320,800 | $722,500 | $1,795,600 | $1,148,100 | $1,450,800 |
| Average cost to delay | $96,139 | $269,599 | $194,081 | $254,761 | $170,453 | $273,234 |
| Average cost to cancellation+ | $17,515 | $72,894 | $56,610 | $91,897 | $54,788 | $76,248 |
| **Total developer cost/project** | **$747,254** | **$1,663,293** | **$973,191** | **$2,142,258** | **$1,373,341** | **$1,800,282** |

An illustrative example of the potential for cost and schedule improvement in a model-based ATO is the Interface Control Document (ICD), which can take several months to produce and may extend to hundreds of pages. Labor savings may be achieved by auto-generating key content of the ICD from a model. In general, these labor savings could be used to lower the costs of the ATO, accelerate the project toward deployment, or improve product quality by enhancing assurance, security, and resiliency. Within the project, a model-enabled ICD can more easily be kept up to date with ongoing system changes, as design modifications are captured in the model and automatically reflected in model-derived subsequent versions of the ICD. For instance, rather than searching through a lengthy ICD to locate interface protocols and data types, the model consistently represents the interfaces, their protocols, and the types of data exchanged among them. Experience with ICDs engineering products indicates that such an approach can reduce labor

hours, or STE, by ~50%, lowering program costs. Similar efficiencies have been demonstrated with many other documents required in the ATO process [82].

The model-based approach enhances the effectiveness of ongoing design analysis and decision-making. For instance, with a focus on security, the model can reveal whether sensitive data is transmitted and stored in ways that may be vulnerable to compromise. It illustrates how data flows through the system and identifies relevant interfaces and protocols, including data encryption methods in transit, in use, and at rest. This understanding is crucial for systems that handle sensitive information, which may have varying storage and transmission security requirements. Specific datasets that traverse a system may necessitate entirely different "landing zones" compared to other datasets in transit simultaneously. Grasping the interfaces and connections is essential to prevent data from being inadvertently accessed or transmitted in unauthorized or undesirable ways. If these interfaces and connections are not fully comprehended, it could lead to a data leak, which may have long-lasting detrimental effects on all stakeholders and jeopardize the system’s long-term viability.

An important aspect of process improvement through a modeling foundation is the reuse of relevant existing content. One clear advantage is the ability to leverage document content from models of systems similar to the ones undergoing an ATO. Furthermore, software-intensive government systems are required to operate within a cloud infrastructure to enhance collaboration and data transparency. This mandate specifies which vendors can provide the necessary infrastructure and services, thereby constraining the variety of tools available for development and deployment—tools that must first undergo accreditation. This approach to software development fosters commonalities among the systems, as they share the same resources.

Consequently, it suggests that a single model can serve as a template for multiple solutions. According to Poissant, the commonality across software systems enables the model to deliver an 80% solution in nearly all scenarios. Reusability thus represents a significant cost-saving opportunity. While MBSE is not a new concept and has seen considerable success within the industry, some organizations face challenges as it requires an initial investment. However, once models are established, they can be utilized throughout the entire lifecycle of the system, often spanning decades [78].

The other benefit of enforcing consistency for security controls across the USG IS model is that the generation of documentation for the ATO process can be performed with automation from a centralized model repository. Referring to Table 5, 41 documents summing to over 4300 pages that are considered candidates for document auto-generation and document replacement directly from the MBSE model of the ATO and USG IS. The documentation highlighted in blue has been identified as candidates for auto-generating documents directly from the USG IS SysML model. The documentation highlighted in red has been identified as candidates to be directly replaceable by the artifacts of the USG IS SysML model. In this implementation, we have used Velocity Template Language (VTL) to script the auto-generation of documents from the SysML model using the tools available in Cameo. The candidacy for template development is due to the frequency with which these documents are used in all USG ISs [54]. When we model the level of effort required to develop these documentations using the metric of the Staff Years of Technical Effort (STE) (commonly used when costing a program, as illustrated in Table 2), the savings in U.S. dollars equate to 24.5% of the ATO documentation cost. Further cost savings can be realized if those documents previously identified in blue are generated for preexisting templates.

## 6.3 METHODS

To test the hypothesis that a model-based ATO can realize the benefits that have been attributed to MBSE in other applications, this dissertation seeks to model both the degree of labor savings available with a model-based ATO and the degree of cost savings available with a model-based ATO. The methods to model these characteristics are presented below.

### 6.3.1 ASSESSING THE EFFECT OF MODEL-BASED TRANSFORMATION ON ATO DOCUMENTATION

Table 5 below presents recommendations for modeling of ATO documentation based on integrating MBSE into the system development and ATO process.

*Table 5 - Documentation recommendation for each document required for ATO.*

| Deprecate | Reduce | Modify | Retain |
|---|---|---|---|
| Software Installation Plan | CONOPS | Software Test Plan (refer to the model) | System Requirements |
| Software Transition Plan | Software Development Plan | Software Requirements Specification (refer to the model – model the requirements) | Security Assessment Report (query the model / logs) |
| System / Subsystem Design Description | Operational Concept Description | Interface Requirements Specification | Security Assessment Plan |
| System / Subsystem Specification | Interface Design Description | Software Version Description to Release Notes | POA&M |
| Software Product Specification | Software Design Description | | Risk Assessment |
| Software User Manual | Software Test Description (mostly automated tests) | | Privacy Impact Assessment |
| Software Center Operations Manual | Software Test Plan | | Privacy Threshold Assessment |
| Software Input / Output Manual | Software Test Reports (daily logs can be parsed) | | Monitor Strategy |
| Computer Operation Manual | System Definition Document | | Continuous Monitoring Plan |
| Computer Programming Manual | System Security Plan (Fortify scans) | | Requirements Traceability Matrix |
| Firmware Support Manual | Status Report (daily logs, automated alerts) | | |
| Disaster Recovery Plan (Availability Zones and cloud solutions prevent this) | Incident Response Plan (software and infrastructure handle most of this) | | |
| ATO Boundary Document (that is the model) | Database Design Description | | |
| Updated System Security Plan | | | |

The column's title represents the recommended action for the type of document listed underneath.

- Eliminate - The data in this document will be captured in the MBSE model or is irrelevant to the example government cloud software system solution.
- Reduce - The model captures most of the information needed to understand the solution, thereby reducing the amount of text in the document by referring to the model.
- Modify - The document briefly describes the information required to perform accreditation and points directly to the model where the solution provides the details.

The following table displays the result of the documentation recommendation from Table 5.

*Table 6 - Result of documentation recommendation*

| Recommendation | Amount |
|---|---|
| Eliminate | 14 |
| Reduce | 13 |
| Modify | 4 |
| Remain | 10* |

This categorization of documents and their potential for change under a model-based ATO scenario provides the potential effort savings available by eliminating documentation or reducing page count.

* Remain unchanged from the document-based ATO baseline.

### 6.3.2 ASSESSING THE EFFECT OF MODEL-BASED TRANSFORMATION ON ATO COSTS

To translate these potentials for time and efforts saved under a model-based ATO process into cost savings we must account for the cost of labor. This variable can have significantly different ranges based on years of experience and geographical location. This research applies a cost per hour based on the general salary for the position of Technical Writer in the Washington D.C. metropolitan area in 2024 ($62.50 per hour, at a publication rate of 1 page per two hours).

Several *assumptions* underlie this analysis, including:

- The STE cost per hour (based on the salary of a Technical Writer in the Northern Virginia Metropolitan Area (a Washington D.C. USA business district))
- A single author
- The overall page count per document (derived from a real-world cloud-centric system development effort)
- The per-page development duration[‡‡]

These are informed assumptions by previous experience with similar development initiatives over several decades. Additionally, it is important to note that this analysis assumes no document revisions, as both draft documents and final documents can be auto-generated from SysML models [83].

## 6.4 RESULTS

The following paragraphs describe the result of chapter six (6) followed by a discussion and conclusion.

[‡‡] Any change to the assumptions will affect the results of this dissertation. These values and assumptions are solely used as examples for this research and dissertation.

### 6.4.1 QUANTIFYING COST SAVINGS THROUGH MBSE

The following paragraphs examine the tangible savings linked to MBSE and the utilization of documentation templates to improve the reusability of common system deliverables. The time savings achieved through the auto-generation of documentation, along with the reduction of unnecessary documents, could lead to a more expedited timeline for deploying mission-critical systems. Nevertheless, various uncontrollable factors discussed throughout this dissertation may impact both the time and cost of system deployment. Furthermore, identifying necessary artifacts for accreditation by the AO can vary significantly based on individual perspectives and specific project requirements, ultimately affecting deployment time and cost.

#### 6.4.1.1 COST ANALYSIS

The table below analyzes the cost implications of an MBSE-enabled ATO within the context of an Asset Management system, focusing specifically on documentation costs related to page count and the assumptions outlined in section 6.3.2. Table 7 provides a detailed breakdown of the typical documentation and reports associated with the ATO. The page count reflects actual data from a smaller, cloud-based transactional application that recently completed the ATO process in 2022.

*Table 7 - Documentation Cost Analysis*

**Sanchez ATO Analysis**

| ATO Step | Diagram References | Non-ATO Documentation – Additional but Recommended – Often Requested | Size/Resources | Comment | ATO Documentation Package - Required | Size/Resources | Common in ATO Package - Additional | | Total Number of Docs |
|---|---|---|---|---|---|---|---|---|---|
| Prepare | | · CONOPS<br>· Software Development Plan<br>· Software Installation Plan<br>· Software Transition Plan<br>· Operational Concept Description<br>· Software Test Plan<br>· System Requirements | 32<br>162<br>15<br>76<br>21<br>40<br>200 | | | | · Privacy Impact Assessment<br>· Privacy Threshold Assessment<br>· Incident Response Plan<br>· Disaster Recovery Plan<br>· ATO Boundary Diagram | | |
| Categorize | | · System / Subsystem Design Description | 436 | | · System Definition Document | 436 | | | |
| Select | | · System / Subsystem Specification<br>· Software Requirements Specification<br>· Interface Requirements Specification<br>· Software Product Specification | 170 | | · System Security Plan | 145 | | | |
| Implement | | · Software Design Description<br>· Interface Design Description<br>· Database Design Description<br>· Software Test Description<br>· Software Test Procedures | 264<br>385<br>15<br>1000 | Generated by the System Model<br>Generated by the System Model<br>Generated by the System Model | · Updated System Security Plan<br>• Status Report | 12 | | | |
| Assess | | · Software Test Report<br>· Software Version Description<br>· Requirements Traceability Matrix | 23<br>2<br>400 | | · Security Assessment Report<br>• Security Assessment Plan | 112<br>63 | | | |
| Authorize | | · Software User Manual<br>· Software Center Operator Manual<br>· Software Input/Output Manual<br>· Computer Operation Manual<br>· Computer Programming Manual<br>· Firmware Support Manual | 172 | | · POA&M<br>· Risk Assessment | 31<br>34 | | | |
| Monitor | | · Continuous Monitoring Plan | 15 | | · Monitor Strategy Document | 10 | | | |
| Page lengths were determined from a true, small, Government cloud software solution. In some cases page numbers are approximations | | | | | | | | | |
| | | Total page count | 3428 | | Total page count | 843 | | | **41** |

**Assume each document is new @ $62.50/hr x 2 hours per page**

| | | | |
|---|---|---|---|
| $428,500 | Total for new Recommended Documents | $105,375 | Total for Required Documents |
| | | $533,875 | Total for Recommended and Required Documents |

**Pages eliminated by MBSE:**

| | |
|---|---|
| Software Installation Plan | 15 |
| Software Transition Plan | 76 |
| System / Subsystem Design Description | 436 |
| ATO Select Step | 170 |
| Software Design Description | 264 |
| Interface & Database Design Description | 385 |
| 1346 | Total Pages |
| $168,250 | Total Savings |
| 31.51% Savings | |

The total savings of $168,250 represents a 31% reduction in documentation costs. This cost analysis serves as a conservative estimate of the potential savings and aims to illustrate the methodology used in the assessment. For the documentation typically required as deliverables in USG IS of this nature, there are conservatively estimated 3,428 total pages. Additionally, around 843 documents pertaining to the Asset Management System are necessary for the ATO. This results in a total estimated cost of $428,500 for a single STE at a rate of $62.50 per hour, assuming two hours of work per page. This is illustrated in Table:

*Table 8 - Cost Analysis Breakdown*

| | |
|---|---|
| Total cost of documentation<br>(Calculated by total number of pages multiplied by assumed cost per-page<br>3428 pages * ($62.50 per hour * 2 hours per-page)) | $428,500 |
| Total savings<br>(Calculated by total number of pages eliminated multiplied by assumed cost per-page<br>1346 pages eliminated * ($62.50 per hour * 2 hours per-page)) | $168,250 |
| Total hours saved<br>1346 pages * 2 hours per page | 2,692 |
| Overall savings percentage | 31% |

In contrast, the anticipated cost for the documents required for the model-based ATO amounts to $105,000. By adopting MBSE, a total of 1,346 pages can be eliminated, leading to total cost savings of $168,250 calculated using the same methodology. These savings are expected to be scalable for cloud-centric Government systems. However, this does not account for the need for revisions. Referring to Figure 30 – Documentation Sequence Diagram, even a small reduction in documentation can positively impact time and cost savings.

By minimizing and eliminating redundant tasks, we can evaluate the cost-effectiveness of measurement by comparing the time saved when delivering the same system to the operational environment with and without using MBSE. It is expected that the application of MBSE, along with a reduced need to define blocks and other model components, will result in significant time savings. This efficiency also extends to implementing security controls, which are now organized in a model format that allows for quick associations with block elements. For example, if the initial system was delivered to operations in 36 months without MBSE, this timeframe serves as a useful baseline. When the opportunity arises to deliver a similar system using MBSE, findings from this dissertation suggest that it could be delivered in a shorter period. This advantage is likely to be applicable to other systems as well, considering that many elements used to model the initial system have already been developed and can be effectively reused.

## 6.5 DISCUSSION

The results of the cost analysis for the MB-ATO process indicate substantial cost savings. However, it is also recognized that considerable effort is necessary to shift to a MBSE development lifecycle, which would enable the realization of these cost savings. This dissertation's cost analysis does not encompass the expenses associated with transitioning to an MBSE program, such as

software licensing, training, and overall preparatory costs. Below, a visual representation illustrates the upfront costs incurred over the duration of the system development project.

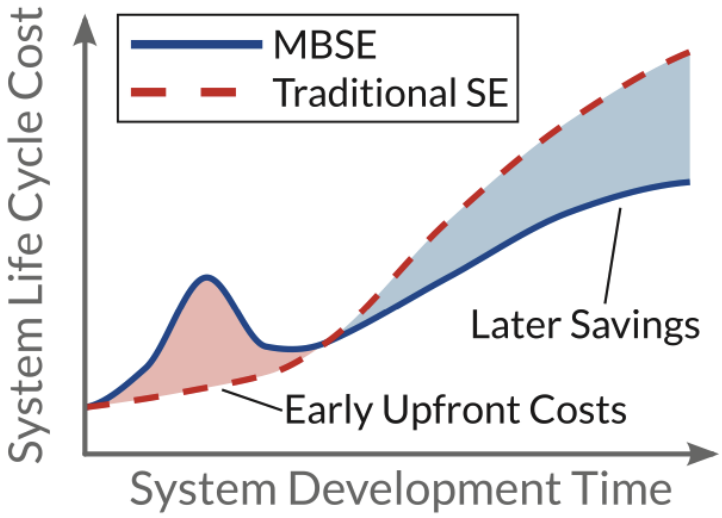



*Figure 39 - SE Cost Over Time for an MBSE Approach Compared to DB-ATO [40]*

Typically, the advantages often attributed to MBSE are not realized early in a system life cycle. The up-front SE cost and effort generally are significantly higher during the early phases of a systems life cycle than those of a traditional DB-ATO approach due to the costs associated with defining an MBSE process, standing up a modeling environment, training staff, configuration management, and doing the actual modeling [40]. The cost model for this study is a model of the "N-th" implementation of the MB-ATO process, where the startup costs of the DBSE to MBSE transition have already been amortized over many previous projects. This cost analysis has demonstrated significant potential for component reuse within the MB-ATO, in which case the startup costs of the digital transformation has been developed as part of this dissertation effort. For example, the MB-ATO process is now available in a SysML format (see Appendix B for additional diagrams), the NIST security controls are available in SysML format (see Appendix E), and the document templates for all ATO deliverables are available in VTL [34]. For these reasons, the

cost analysis's assumption that the transition costs for MB-ATO can be expected to be small is justified.

The documents that are the autogenerated outputs of the MB-ATO process must be validated and human-checked before they are included in the final ATO submission packages. The cost analysis performed in this research does not include additional cost that might be associated with human validation of machine-generated documents [84]. This research asserts that one of the benefits of MB-ATO is that because the ATO-submission documents are derived directly from the IS model, the quality, consistency, and content of the documents are directly derived from the quality, consistency, and content of the model [17]. Although there will be final quality checks, final validation checks, and final documentation assembly required for the MB-ATO documents, these same processes are required for the DB-ATO documents. Therefore, both processes require effort associated with final checking, and these make a negligible contribution to total project costs. However, it is expected that the final review and validation checks required for the documentation produced from the machine-generated process will require less time so long as the model is accurate to date.

This cost analysis demonstrates significant cost reductions by reducing the efforts associated with technical writing. Still, the digital transformation of the organization will require a revision of the types of labor and, therefore, labor costs that they encumber. For example, the reduction in technical writing effort may be replaced by an increase in technical modeling effort.

While the cost analysis for this research cannot fully account for the changes in labor, roles, and capabilities that organizations will experience during digital transformation, it does illustrate the types of labor and costs that can be reallocated to achieve greater organizational efficiencies.

In summary, although there are upfront costs associated with transitioning to an MBSE development lifecycle, evidence suggests that there are long-term benefits and a reduction in overall costs. Utilizing machine-generated documents from model templates allows for quicker documentation processes and enhances consistency, provided that the model is kept current and accurate. Additionally, the validation process for these documents before signatures will likely require less time.

Lastly, measuring the exact labor, roles, and capability changes can be challenging, given the uniqueness of each project and organization. However, it is reasonable to anticipate a redistribution of efforts and roles within the organization.

## 6.6 CONCLUSION

This chapter presents an introduction, methods, and results demonstrating a significant opportunity to quantify the total savings realized through MBSE in the government sector. MBSE can expedite the deployment of systems into the operational environment while simultaneously enhancing quality and reducing costs. By allocating resources to conduct a real-world exercise of "fielding" a sample system both with and without the application of MBSE within the government arena, a comprehensive analysis of the time, cost, and resources necessary to achieve ATO can be conducted. This dissertation establishes that this endeavor is feasible.

The decision to invest in this opportunity ultimately rests with the U.S. Government. It is recommended that funding be allocated for research that explores both the advantages and disadvantages of this methodology within a system that could be operationally implemented. Such research would provide a practical model for similar future projects and establish a baseline for subsequent development efforts. Once a benchmark has been established, a design of experiments can be conducted to evaluate the impact of changes associated with specific variables. For instance,

if the benchmark is developed with a team of highly skilled system modelers well-versed in tools such as Cameo Systems Modeler or Cameo Magic Systems of Systems Architect, as referenced interchangeably in this dissertation, an experiment could then investigate the implications of executing a project without such skilled personnel and the necessary training required to bring them up to speed. Training costs can be recorded for the system development effort and used to create more accurate cost assumptions for future projects lacking that specific skill set.

# CHAPTER 7 – SUMMARY

This section summarizes the unique research approach used to address the problems outlined in Chapter one (1) and the model-based solutions and alternatives to the ATO process. This final chapter includes the synthesis of the results gathered from the chapters, conclusions derived from the research questions and tasks, a preliminary validation of the work presented in this dissertation, and recommendations for future research on the subject.

## 7.1 SYNTHESIS OF RESULTS

Implementing an MBSE approach to Government IS security accreditation are demonstrated in this research to yield measurable improvements in a variety of metrics of accreditation performance. This dissertation provides research that shows the impact of MBSE on system accreditation and development by reducing inconsistencies, promoting reuse, and reducing the time to deploy to the operational environment while potentially leading to cost savings. The summation of this dissertation is as follows.

Chapter three (3) outlined the approach taken to model the ATO and presented an example of an Asset Management system, also modeled using MBSE tools and SysML. These models effectively capture the expectations detailed in the Government IS development framework and provide a basis for subsequent analysis. They reflect a practical interpretation of what is commonly seen in cloud-centric systems within the U.S. Government domain.

Chapter four (4) examined how these models can address the inconsistencies typically encountered in traditional methodologies or the DB-ATO. The Model-Based Structured Requirements demonstrated forward and backward traceability, linking requirements to

corresponding test cases and procedures. The supporting diagrams also reinforced this traceability throughout the process, from requirements to test cases and procedures.

Chapter five (5) showcased the advantages of modeling for reuse. It tackled the challenges associated with the copy-and-paste approach, proposing an object-oriented strategy for reuse as an alternative. This chapter emphasizes how sound design, and architecture can enhance accreditation when reused effectively. For the first time, it introduced a Sequence Diagram illustrating ATO documentation by process steps of the RMF. Furthermore, it explored the application of templates for generating system documents, aiming to minimize the manual effort traditionally involved in the DB-ATO process. Examples were provided to illustrate the portability of custom elements for reuse across similar projects.

Chapter six (6) explored the potential for accelerating the release of production-ready systems into the operational environment by utilizing the methodologies outlined in chapters 3, 4, and 5. While acquiring empirical data can be notoriously difficult [80] [40], the findings presented in chapter six (6) indicate significant benefits in incorporating MBSE into the ATO process. To date, there has not been a counterargument to dispute the idea that MBSE is beneficial, just that the quantification from evidence is lacking [40]. This provides compelling evidence that modernizing the ATO through MBSE can lead to faster releases at a lower cost, which is crucial for maintaining competitiveness on the global stage and ensuring the success of end-users.

## 7.2 CONCLUSIONS DERIVED

Quantifying the improvements gained from applying MBSE to the ATO process presents challenges, as no two development efforts are identical. This is true even for systems that deliver similar functionalities, utilize the same tools and technologies, and operate within the same environment. A variety of factors contribute to their uniqueness, with one often-overlooked aspect

being the personnel and supporting data required by an AO. Additionally, the task is complicated by the difficulty in obtaining foundational timelines, such as development and deployment schedules, as well as cost data. Nevertheless, the information gathered from this research indicates that implementing MBSE for system accreditation yields positive outcomes, particularly for similar systems and those that are notably complex and, or have extended lifecycles.

A study by the Aerospace Corporation, a Federally Funded Research and Development Center (FFRDC) determined that in a 2023 cost study of MBSE for the NASA Cost and Schedule Symposium that "MBSE cost data is rare with no current enterprise-wide implementation [80]." Their findings conclude that:

- Little information for costing analogies for MBSE on large programs
- Actual vs Theory: Effectiveness is dependent on systems engineers
- Utilizing the tools effectively, tools cannot take the place of systems engineers (cost factor to skills of the systems engineers)
- MBSE culture and experience of execution inside of a company is a good indicator of efficient MBSE
- MBSE must be utilized throughout the System Engineering Life Cycle (SELC) of a program to be effective – configuration management must be sustained (and cost for)
- MBSE tools can't replace good systems engineering (watch item for contracts)
- MBSE process is ever-changing and is not standardized yet across the government – costing is similar to paying for programmers at this time

*"Evidence points to but cannot confirm MBSE costs up front reduce risk and standardize architectures across a program* [80]*"*

Arguably the most important factor in increasing the perceived compatibility of MBSE is to emphasize its compatibility with the pressing needs of systems engineers. As pointed out in Section 1.4.1, where challenges were presented with traceability and consistency with the traditional DB-ATO, a need that MBSE is particularly suited to address is the difficulty in maintaining the consistency of Systems Engineering artifacts. As presented earlier, this is particularly challenging in the presence of requirement, constraint, and design changes that are inevitable in modern, complex systems. Maintaining consistency in a document-based approach can be challenging because there is no automated method to propagate changes made to a system element across all references in various locations, such as text, figures, tables, and matrices within multiple documents. When a change occurs, it must be manually updated in every instance, which increases the risk of input errors or missing some references that need to be revised. Such mistakes can be costly, as engineering teams in other domains such as electrical, mechanical, and software engineering—rely on specifications generated by systems engineers. Any inconsistencies or errors in these specifications could adversely affect their designs [40].

In contrast, a well-designed MBSE approach ensures that any change made to product information within the model is automatically updated wherever that element is referenced. Additionally, MBSE allows for integrating other product information repositories, such as product lifecycle management tools or requirements databases, with the system model. The ability to manage change effectively and maintain consistency is a compelling advantage that aligns well with the principles of MBSE [40].

MBSE has the potential to significantly streamline security documentation, condensing volumes of material from hundreds or even thousands of pages to a more manageable format through the implementation of digital engineering practices, specifically MBSE and SysML.

Although comprehensive information regarding the explicit acceleration of the ATO process through MBSE is somewhat scarce, it is evident that this approach presents a promising means of reducing the deployment time for mission-critical systems.

The accreditation of U.S. government information systems does not adhere to a single uniform methodology, as each system possesses unique characteristics in its services and associated security requirements. Nonetheless, the RMF delineates crucial steps necessary to ensure that a system obtains accreditation and becomes operational. Securing an ATO is not simply a beneficial advantage; it is an essential requirement for the successful delivery of projects.

The ATO delineates a flexible framework by outlining the high-level accreditation requirements that must be met. It is the responsibility of the development team to ensure that these requirements are satisfied, ideally on the first attempt. This process can be rendered significantly more efficient when security considerations are integrated into the development team's efforts from the initial stages, continuing seamlessly into operational phases.

Despite the challenges associated with resource availability, particularly recruiting qualified personnel, numerous tools exist to streamline and automate many routine tasks. Such automation has the capacity to mitigate inconsistencies and enhance the quality of the overall system. MBSE can provide valuable insights into both the nature and physical architecture of systems, facilitating the rapid identification of potential vulnerabilities, such as open ports, interfaces, and third-party components.

There exists a concern regarding the reliance on a singular decision-maker for the authorization process, which may not always be advisable. Research conducted for this dissertation suggests that MBSE could empower additional decision-makers to authorize systems in situations of critical necessity without the immediate involvement of the designated AO.

Documentation often demands considerable time and is particularly susceptible to errors and inconsistencies, especially when cross-referencing is needed for traceability purposes. Dependence on simplistic copy-and-paste techniques for documentation is fraught with challenges. However, using MBSE diagrams can markedly decrease the volume of documentation required to convey the same information effectively.

## 7.3 RESEARCH CONTRIBUTIONS

Having completed this set of research projects, the contributions of this dissertation can be summarized as follows:

This dissertation introduces a model-based process for the ATO and RMF procedures currently required for accrediting USG ISs. It represents the first SysML model-based ATO process documented in the open literature.

Additionally, this dissertation has developed a SysML model of a representative USG IS and demonstrated the process by which this system would undergo the model-based ATO process for accreditation. This dissertation clearly defines the key characteristics of the IS model that drive the performance, cost efficiency, and reusability benefits inherently associated with MBSE processes. The IS model features structured model-based requirements, automated document generation, and reusable libraries of NIST SP-800 security control models, all of which are directly traceable to quantified performance improvements (enhanced requirements traceability and consistency), cost reductions (a modeled decrease of 31% in this example), and significant reusability advantages (over one thousand of the NIST Security and Privacy controls available for reuse in SysML).

The findings of this dissertation provide compelling evidence that the processes for approving and accrediting USG ISs can be innovated through technological and procedural advancements. It is essential for the accreditation field to invest time and resources in researching

and prototyping innovative approaches, moving away from outdated methods and excessive documentation toward a more modern, streamlined, and digital methodology. This dissertation has demonstrated that systems engineering methods, particularly MBSE-enabled ATO, can significantly enhance the quality and cost-effectiveness of the accreditation process.

The results of this research are currently under review for publication in a peer-reviewed journal and have been accepted and presented at peer-reviewed conferences.

## 7.4 RECOMMENDATION FOR FUTURE RESEARCH

It would be advantageous for the U.S. Government to embark on research initiatives that explore how MBSE can facilitate a more efficient ATO process. Such research would enhance the existing body of knowledge and provide valuable insights into the advantages and disadvantages of adopting MBSE methodologies. By examining the costs involved in initiating a new project versus integrating MBSE into an existing one, we can better understand the financial implications and derive significant value. This research could serve as a framework for future development projects.

MBSE holds promises for success, and to maintain a competitive edge, it is crucial to investigate contemporary methodologies. Additionally, there is substantial potential for further analysis regarding the application of Artificial Intelligence (AI) within the ATO process. The integration of AI and MBSE may yield even greater efficiencies and benefits. Initial research indicates the availability of white papers and trade literature that address the implications of AI on application security:

- The Department of Defense Releases AI Adoption Strategy,
- Advancing Governance, Innovation, and Risk Management for Agency Use of Artificial Intelligence,

- The Department of Health and Human Services Looks to AI as a Tool for Speeding the Systems Authorization Process.

The widespread adoption of AI in the commercial sector suggests that its implementation within U.S. Government Information Systems is only a matter of time. This promising potential for AI integration brings forth several challenges that require careful examination, including the data sources used for machine learning, data integrity issues, and security implications. The machine learning component introduces additional complexities, such as transferring data from unclassified "low-side" networks to classified "high-side" environments. While maintaining data integrity is paramount, the potential advantages of this integration are significant and could transform the ATO process.

Creating and analyzing activity elements within Cameo, or equivalent modeling software, for each of the over one thousand security controls would significantly enhance the risk analysis capabilities for simulations concerning the system's defensive mechanisms and alerting posture. By integrating security controls into the model that closely reflects modern technical defenses, we can establish assertions as minimum and maximum values, which can then be manipulated across various threat scenarios. The outcomes of these simulations would closely simulate how the system might perform under stress, effectively mitigating brute force attacks and other malicious activities. Furthermore, these activities can be utilized in the modeled ATO environment to conduct realistic simulations for future systems seeking accreditation, ideally in an MBSE format.

## 7.5 PRELIMINARY VALIDATION

This work has undergone a thorough review and is currently in the peer-review process for publication in the INCOSE Systems Engineering Journal[8]. The chapters of this dissertation are closely aligned with real-world experiences, which have been emphasized where applicable. Several senior cybersecurity experts from the U.S. Department of Defense, including representatives from government service, Federally Funded Research and Development Centers (FFRDC), and commercial contractors were consulted and interviewed to evaluate this methodology. All experts hold senior technical or managerial positions and possess relevant certifications in the cybersecurity field, including a CISSP, ensuring a thorough review process. Furthermore, there has been expressed interest within the U.S. Government, which has indicated that there is potential for achieving these improvements and that opportunities exist for securing funding for further research in the governmental sector. Additionally, this dissertation was developed under the leadership, experience, and guidance of the Colorado State University advising committee in the Systems Engineering Department.

## 7.6 DISCLAIMER

The views expressed in this dissertation and the component research questions and tasks (case studies) are solely the author's and do not represent the position of the U.S. Government, the U.S. Department of Defense, the U.S. Intelligence Community, Colorado State University, or any other public or private organizations.

[8] "A Model-Based Approach to Achieve Authorization to Operate by Reducing Inconsistences" was submitted to the Systems Engineering Journal of INCOSE – submission ID 2c4a1d3d-06f4-4235-b2bd-815f5962a7e1 – and is awaiting publication.

# REFERENCES


[1] Department of Defense Standard, *Department of Defense Trusted Computer System Evaluation Criteria,* Department of Defense, 1985.

[2] J. L. Valladares, "Effectiveness of the Department of Defense Information Assurance Accreditation Process," U.S. Army War College, Philadelphia, 2013.

[3] D. de Zafra, S. i. Pitcher, J. D. Tressler and J. B. Ippolito, "Information Technology Security Training Requirements A Role and Performance-Based Model," NIST SP 800-16, Gaithersburg, 2021.

[4] T. A. Chick, "Maintaining Your Authority to Operate (ATO) While Being Agile: Achieving Continuous Reauthorization with DevOps," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2018.

[5] M. E. Whitman and H. J. Mattord, Principles of Information Security, 6th Edition, Cengage Learning, 2014.

[6] J. A. Bennerson, "Navigating the US Federal Government Agency ATO Process for IT Security Professionals," *ISACA,* 2017.

[7] NIST, "About the Risk Management Framework - A Comprehensive, Flexible, Risk-Based Approach," 2024. [Online]. Available: https://csrc.nist.gov/projects/risk-management/about-rmf.

[8] T. Chick and T. Scanlon, "Risk Management Framework (RMF) and Authority to Operate (ATO)," Software Engineering Institute - Carnegie Mellon University, Pittsburgh, 2023.

[9] M. Cribbs, "What is an Authority to Operate?," 2002. [Online]. Available: https://www.secondfront.com/insights/what-is-an-authority-to-operate-ato.

[10] NIST, "NIST," 2023. [Online]. Available: https://csrc.nist.gov/Projects/risk-management/sp800-53-controls/release-search#/controls?version=5.1.

[11] G. Nemr, "Obtaining Information Assurance (IA) accreditation for systems initially deployed without ia considerations," in *IEEE Military Communications Conference*, San Diego, 2008.

[12] M. J. Cotteleer, S. S. Goldenberg, I. Wing, O. Alliyu, S. Kania, V. Mujumdar and B. Sinderman, "ACM Digital Library," 2021. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3462223.3485624.

[13] C. Gunderson, "Naval Postgraduate School," 2014. [Online]. Available: https://calhoun.nps.edu/handle/10945/43220.

[14] S. Bocetta, "DevOps," 2020. [Online]. Available: https://devops.com/devsecops-vs-agile-development-putting-security-at-the-heart-of-program-development/.

[15] S. Pendino, R. K. Jahn and K. Pedersen, "U.S. Cyber Deterrence: Bringing Offensive Capabilities into the Light," *Joint Force Staff College - Academic Journals,* 2022.

[16] L. A. Odell, C. E. DePuy, J. C. Fauntleroy, T. C. Rabren and M. G. Seitz-McLeese, "Recommendations for Improving Agility in Risk Management for Urgent and Emerging Capability Acquisitions — Quick Look Report," Institute for Defense Analyses, 2017.

[17] J. M. Borky, *Monitor Security Controls,* Fort Collins, CO: Colorado State University - Course Material, 2015.

[18] NIST Risk Management Framework, "About the Risk Management Framework," 25 July 2024. [Online]. Available: https://csrc.nist.gov/projects/risk-management/about-rmf.

[19] L. Stanton, "GSA Blogs," 2020. [Online]. Available: https://gsablogs.gsa.gov/technology/2020/10/30/authorization-to-operate-preparing-your-agencys-information-system/.

[20] Office of the Under Secretary of Defense for Research and Engineering, *Systems Engineering Guidebook,* Washington DC: Office of the Deputy Director for Engineering, 2022.

[21] INCOSE, "(Brief) History of Systems Engineering," 2024. [Online]. Available: https://www.incose.org/about-systems-engineering/history-of-systems-engineering.

[22] E. C. Honour, "Systems Engineering Return on Investment (Thesis PhD)," University of South Australia, 2013.

[23] W. F. Frantz, "The Impact of Systems Engineering on Quality and Schedule * Empirical Evidence," *INCOSE International Symposium,* vol. 5, no. 1, pp. 618-624, 1995.

[24] E. R. Carroll and R. J. Malins, "Systematic Literature Review: How is ModelBased Systems Engineering Justified?," Sandia National Laboratories, Albuquerque, 2016.

[25] D. J. Shepard and J. Scherb, "What is Digital Engineering and How Is It Related to DevSecOps?," 2020. [Online]. Available: https://insights.sei.cmu.edu/blog/what-digital-engineering-and-how-it-related-devsecops/.

[26] Idaho National Labs, "Digital Engineering," 2024. [Online]. Available: https://inl.gov/digital-engineering/#:~:text=Digital%20engineering%20describes%20a%20holistic,on%20construction%20cost%20and%20schedule..

[27] FedScoop, "How Digital Engineering Models are Changing IT Development," FedScoop Report, 2023.

[28] DoD Instruction 5000.97, "DoD Instruction 5000.97 Digital Engineering," Office of the Under Secretary of Defense for Research and Engineering, 2023.

[29] D. Hettema, "Overview of Digital Engineering, Modeling & Simulation for DAU SE Modernization," Under Secretary of Defense for Research and Engineering, 2022.

[30] Prostep, "Digital Engineering vs. MBSE: What Are the Main Differences Between the Two?," 30 September 2024. [Online]. Available: https://prostep.us/blog/digital-engineering-vs-mbse-what-are-the-main-differences-between-the-two/#:~:text=DE%20provides%20the%20concepts%20of,stage%20of%20the%20development%20lifecycle..

[31] C. R. China, "Digital twin vs. digital thread: Two complementary ways to digitally replicate assets," 2023. [Online]. Available: https://www.ibm.com/think/topics/digital-thread-vs-digital-twin.

[32] J. M. Borky and T. H. Bradley, "Effective Model Based Systems Engineering," Cham, Switzerland, Springer, 2019.

[33] P. Younse, J. Cameron and T. H. Bradley, "Comparative Analysis of Model-Based and Traditional Systems Engineering Approaches for Simulating a Robotic Space System Architecture Through Automatic Knowledge Processing," *Systems Engineering,* pp. 360-384, 2022.

[34] Dassault Systemes, "Velocity Templating Language," 2024. [Online]. Available: https://docs.nomagic.com/display/MD190/Velocity+templating+language.

[35] K. Henderson and A. Salado, "Value and Benefits of Model-Based Systems Engineering (MBSE): Evidence from the Literature," *Systems Engineering,* pp. 24: 51-66, 2021.

[36] L. E. Hart, "Introduction to Model-Based Systems Engineering (MBSE) and SysML," in *Delaware Valley INCOSE Chapter Meeting*, 2025.

[37] T. A. Chick, "Are your DevSecOps Capabilities Mature?," Carnegie Mellon University, Pittsburgh, 2023.

[38] G. E. Box, "Robustness in the Strategy of Scientific Model Building," *Robustness in Statistics,* p. 40, 01 April 1979.

[39] P. Els, "Moded-Based Sysstems Save Development Time and Money," 2019. [Online]. Available: https://www.automotive-iq.com/autonomous-drive/columns/model-based-systems-save-development-time-and-money.

[40] D. R. Call, S. Conrad and D. R. Herber., "The effects of the assessed perceptions of MBSE on adoption," in *INCOSE 2024 International Symposium*, Dublin, 2024.

[41] SysML.org, "SysML Open Source Project," 2024. [Online]. Available: https://sysml.org/sysml-faq/what-is-sysml.html.

[42] S. Friedenthal, A. Moore and R. Steiner, A Practical Guide to SysML - The Systems Modeling Language (3rd Edition), Waltham: Morgan Kaufmann, 2015.

[43] Cameo Magic Solution, "Behavioral Diagrams: Use Case Diagrams," 2024. [Online]. Available: https://www.cameomagic.com/language-tool-guidance/behavioral-diagrams/use-case-diagrams.

[44] Department of Defense, "DoD Cloud Strategy," 2018. [Online]. Available: https://media.defense.gov/2019/Feb/04/2002085866/-1/-1/1/DOD-CLOUD-STRATEGY.PDF.

[45] D. R. Herber, J. B. Narsinghani and K. Eftekhari-Shahroudi, "Model-Based Structured Requirements in SysML," *IEEE 2022 International Systems Conference,* 2022.

[46] J. Peacock, "NIST SP-800-53 Control Families Explained," CyberSaint Securit, 2024.

[47] D. R. Herber and K. Eftekhari-Shahroudi, "Building a Requirements Digital Thread from Concept to Testing Using Model-Based Structured Requirements Applied to Thrust Reverser Actuation System Development," *Recent Advances in Aerospace Actuation Systems and Components,* 2023.

[48] R. S. Carson, "Implementing Structured Requirements to Improve Requirements Quality," *INCOSE International Symposium,* vol. 25, no. https://doi.org/10.1002/j.2334-5837.2015.00048.x, pp. 54-67, 2015.

[49] R. S. Carson and R. A. Noel, "Formalizing Requirements Verification and Validation," *INCOSE International Symposium,* vol. 28, no. https://doi.org/10.1002/j.2334-5837.2018.00517.x, pp. 805-818, 2018.

[50] E. C. D. Albornoz-Braojos, "Developing Airplane Systems Faster and with Higher Quality through Model-Based Engineering," *Innovation Quarterly,* pp. 38-40, 01 05 2017.

[51] E. Ladzinski and F. Popielas, "Simplifying Model-Based Systems Engineering - An Implementation Journey," 2024. [Online]. Available: https://discover.3ds.com/sites/default/files/2020-05/simplifying-model-based-systems-engineering-white-paper.pdf.

[52] R. Carson, "Using Architecture and MBSE to Develop Validated Requirements". United States of America Patent 8,886,588, 2020.

[53] P. Els, "iQ Automotive," 2019. [Online]. Available: https://www.automotive-iq.com/autonomous-drive/columns/model-based-systems-save-development-time-and-money.

[54] Leslie, "The Modern Analyst," 2010. [Online]. Available: https://www.modernanalyst.com/Community/CommunityBlog/tabid/182/articleType/ArticleView/articleId/1270/Best-Practices-Inconsistent-Documentation.aspx.

[55] D. Long, "The Fool's Errand of Reuse in MBSE," 2016. [Online]. Available: https://systems-wise.com/the-fools-errand-of-reuse-in-mbse/.

[56] W. Alexander and R. Schiller, "Learn Python Basics," 2024. [Online]. Available: https://openclassrooms.com/en/courses/6902811-learn-python-basics/7091081-import-python-libraries.

[57] E. Gamma, R. Helm, R. Johnson and J. Vlissides, Design Patterns - Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.

[58] ECRI Institute, "Copy/Paste: Prevalence, Problems, and Best Practices," ECRI Institute, 2015.

[59] P. French, "Copy/Paste: “A Computer on Every Desk, and in Every Home, Running Microsoft Software.”," 2024. [Online]. Available: https://www.cardinalpeak.com/blog/copy-paste-a-computer-on-every-desk-and-in-every-home-running-microsoft-software.

[60] Q. Wu, D. Gouyon, É. Levrat and S. Boudau, "Use of Patterns for Know-How Reuse in a Model-Based Systems Engineering Framework," *IEEE Systems Journal, vol.14, no. 4,* pp. 4765-4776, 2020.

[61] I. Dahl, "From MBSE Models to Generated Documents," 2021. [Online]. Available: https://www.samares-engineering.com/en/2021/02/19/from-mbse-models-to-generated-documents/.

[62] IBM, "Cyclomatic Complexity," 2021. [Online]. Available: https://www.ibm.com/docs/en/raa/6.1?topic=metrics-cyclomatic-complexity.

[63] J. M. Bieman, "Reuse Metrics for Object Oriented Software," NASA Langley Research Center, Fort Collins, CO, 1998.

[64] W. Ary, Interviewee, *Experience with the ATO.* [Interview]. 2024.

[65] A. M. Madni and S. Purohit, "Economic Analysis of Model-Based Systems Engineering," *Systems,* 2019.

[66] A. E. Trujillo and A. M. Madni, "MBSE Methods for Inheritance and Design Reuse," in *Handbook of Model-Based Systems Engineering*, Cham, Springer, 2020.

[67] B. A. Nardi, "A Small Matter of Programming: Perspectives on End-User Computing," MIT Press, Cambridge, 1993.

[68] C. Perrone and A. Repenning, "Graphical rewrite rule analogies: avoiding the inherit or copy and paste reuse dilemma," *IEEE Symposium on Visual Languages (Cat. No.98TB100254),* pp. 40-46, 1998.

[69] K. Stemmler, "4 Principles of Object-Oriented Programming," 2022. [Online]. Available: https://khalilstemmler.com/articles/object-oriented/programming/4-principles/.

[70] S. Chatterjee, "What is Object-Oriented Programming and Why is it Useful?," 2024. [Online]. Available: https://emeritus.org/blog/coding-what-is-object-oriented-programming/#advantages-of-object-oriented-programming.

[71] P. Mohagheghi and R. Conradi, "An empirical investigation of software reuse benefits in a large telecom product," *ACM Transactions on So□ware Engineering and Methodology (TOSEM),* pp. 1-31, 2008.

[72] U. Shani and H. Broodney, "Reuse in model-based systems engineering," *2015 Annual IEEE Systems Conference (SysCon) Proceedings,* pp. 77-83, 2015.

[73] National Institute of Standards and Technology Special Publication 800-37, Revision 2, "National Institute of Standards and Technology Special Publication 800-37, Revision 2," Department of Commerce, 2018.

[74] NIST, "NIST Special Publication 800-53 Rev. 5," US Department of Commerce, Gaithersburg, 2020.

[75] Y. Kosarenko, "How to Manage, Share, and Reuse Requirements," [Online]. Available: https://www.ewsolutions.com/how-to-manage-share-and-reuse-requirements/.

[76] M. Chodas, "Improving the Design Process of the REgolith Imaging X-ray Spectrometer (REXIS) Using Model-Based Systems Engineering (MBSE)," in *NASA GSFC Systems Engineering Seminar*, 2014.

[77] S. Kent, "From Cloud First to Cloud Smart," Federal Cloud Computing Strategy, [Online]. Available: https://cloud.cio.gov/strategy/#fedramp. [Accessed 15 10 2024].

[78] A. Poissant, "Agile Model-Based Systems Engineering 71% Faster Than Document-Based Approach," 2021. [Online]. Available: rite-solutions.com/agile-mbse-faster-than-document-based-methodology.

[79] M. Huss, D. R. Herber and J. M. Borky, "Comparing Measured Agile Software Development Metrics Using an Agile Model-Based Software Engineering Approach versus Scrum Only," *MDPI Software Journal,* pp. 310-331, 2023.

[80] B. Cavell and K. Lam, "Model-Based Systems Engineering Cost Study," The Aerospace Corporation, Chantilly, VA, 2023.

[81] S. L. Dawson, A. Batchelor, D. Arenson, J. Adams, S. Simske and D. Wise, *Determining systems engineering value in competitive bids,* Fort Collins: Mountain Scholar Digital Collections of Colorado, 2023.

[82] J. Krasner, "How Product Development Organizations can Achieve Long-Term Cost Savings Using Model-Based Systems Engineering (MBSE)," Embedded Market Forecasters, 2015.

[83] I. Rountree, "MBSE Applications for the MSR SRC Mars Ascent Vehicle," IEEE, Woodbridge, 2022.

[84] J. L. A. Alvarado Jr. and T. H. Bradley, "Developing Model-Based Flight Test Scenarios," *The ITES Journal of Test and Evaluation,* p. Volume 44 Issue 4, 2023.

[85] R. J. M. Edward R. Carroll, "Systematic Literature Review: How is Model-Based Systems Engineering Justified?," Sandia National Laboratories, Albuquerque, 2016.

[86] Visure, "Traditional Systems Engineering Vs Model-Based Systems Engineering (MBSE)," 13 04 2024. [Online]. Available: https://visuresolutions.com/mbse-guide/traditional-vs-modern-system-engineering/.

[87] CISO, "CMS," [Online]. Available: https://security.cms.gov/learn/authorization-operate-ato.

[88] Alpha Omega Integration, "Authorization to Operate (ATO) and Risk Management Framework (RMF) — Overview and Challenges," 2023. [Online]. Available: https://alphaomega.com/blog-post/authorization-to-operate-ato-and-risk-management-framework-rmf-overview-and-challenges/.

[89] D. D. Walden, G. J. Roedler, K. Forsberg, R. D. Hamelin and T. M. Shortell, Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, Hoboken: John Wiley & Sons Inc., 2015.

[90] E. Frenandez, R. Monge and K. Hashizume, "Building a security reference architecture for cloud systems," *Requirements Eng,* pp. 225-249, 06 January 2016.

[91] R. K. Crain, "MBSE without a Process-Based Data Architecture is just a random set of Characters," IEEE, Houston, 2014.

[92] M. E. Gooden, "Return on Investment for Complex Projects Utilizing Model-Based Systems Engineering," George Washington University, Washington DC, 2016.

[93] J. P. Lerat, "Three Reasons Why Document-Based (usually) works better than (most of) MBSE," in *INCOSE International Symposium*, Chicago, 2010.

[94] M. LaSorda, "Applying Model-Based Systems Engineering to Architecture Optimization and Selection During Systems Acquisition," Colorado State University, Fort Collins, 2018.

[95] E. MacAskill, "The Guardian," 2012. [Online]. Available: https://www.theguardian.com/world/2012/apr/22/iran-reverse-engineer-spy-drone.

[96] "Defense Counterintelligence and Security Agency Assessment and Authorization Process Manual," 2020. [Online]. Available: https://www.dcsa.mil/Portals/91/Documents/CTP/tools/DCSA_Assessment_and_%20Authorization_Process_Manual_Version_2.1.pdf.

[97] A. Zacharias, "The ultimate guide to process documentation (with templates)," 2023. [Online]. Available: https://www.notion.so/blog/process-documentation-template.

[98] IcePanel, "Comparison: C4 Modelling vs. Diagramming," 2024. [Online]. Available: https://icepanel.medium.com/comparison-c4-modelling-vs-diagramming-1e51c839630e.

[99] "From MBSE Models to Generated Documents," 2021. [Online]. Available: https://www.samares-engineering.com/en/2021/02/19/from-mbse-models-to-generated-documents/.

# APPENDIX A – SYSTEM ELEMENT SPECIFICATION

An example of a System Element Specification for the Project Staff block in Figure 16.

*Table 9 - System Element Specification - Project Staff*

| **Domain: Project Staff** |
|---|
| |
| General: This Domain contains the system's personnel required to build and deliver a system successfully. These people are responsible for understanding the needs, requirements, and tools and provide experience in areas not limited to management and engineering to meet the demands of the customer and end user. |
| |
| ***Definitions***: see Integrated Dictionary. |
| |
| ***Operations***: |
| • Product Ownership |
| • Program Management |
| • Software Development |
| • Scrum Management |
| • Test |
| • Systems Engineering |
| • Software Engineering |
| |
| ***Values****:* |
| • Staff |
| • Schedule |
| • Program Manager |
| • Lead Software Developer |
| • Lead Software Developer |
| • Scrum Master |
| • Test Engineer |
| • Test Lead |
| • Lead Systems Engineer |
| • Systems Engineer |
| • Software Engineer |
| • Jr. Software Engineer |
| • Project Owner |
| *Allocated Requirements*: *As written in the Statement of Work (SoW).* |

# APPENDIX B – ADDITIONAL MODEL DIAGRAMS

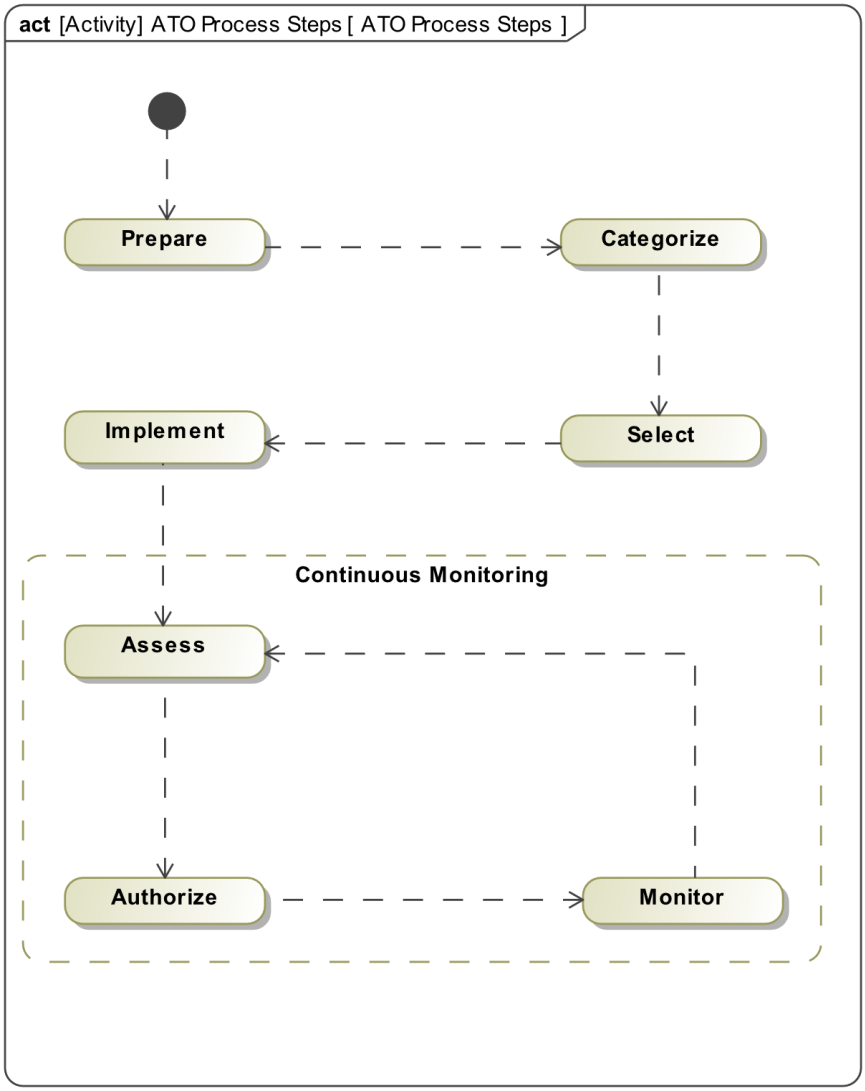



*Figure 40 - Activity Diagram – ATO Process Steps*

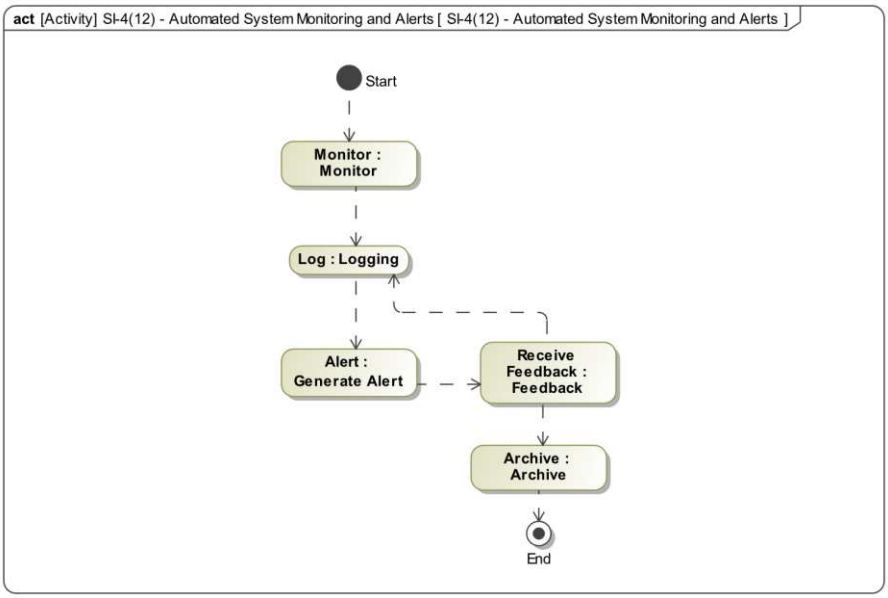



*Figure 41 - NIST 800-53 Rev 5 SI-4(12) Security Control Activity Diagram*

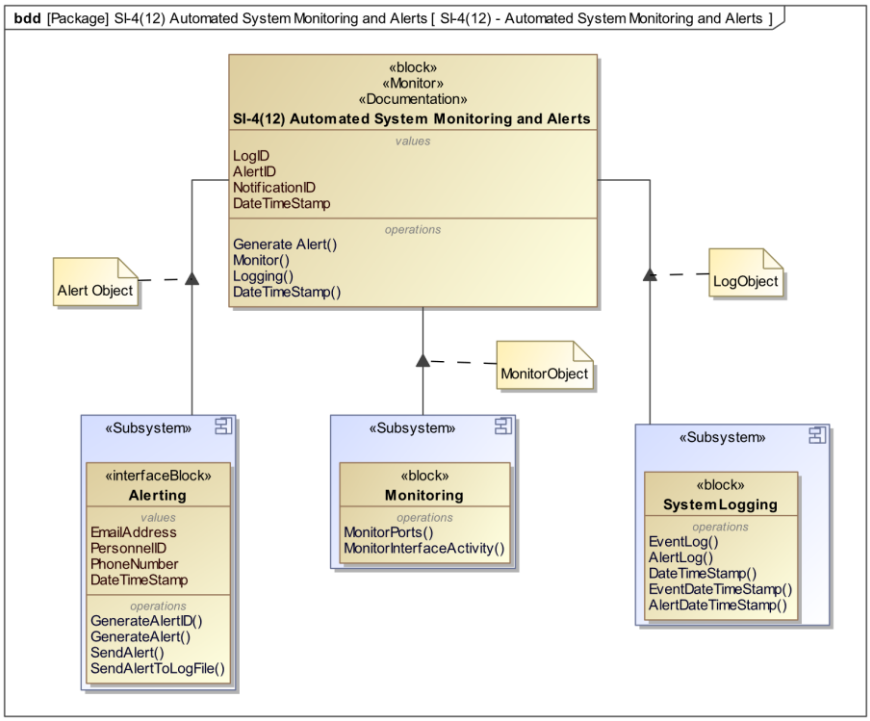



*Figure 42 - NIST 800-53 Rev 5 SI-4(12) Security Control Structure Diagram*

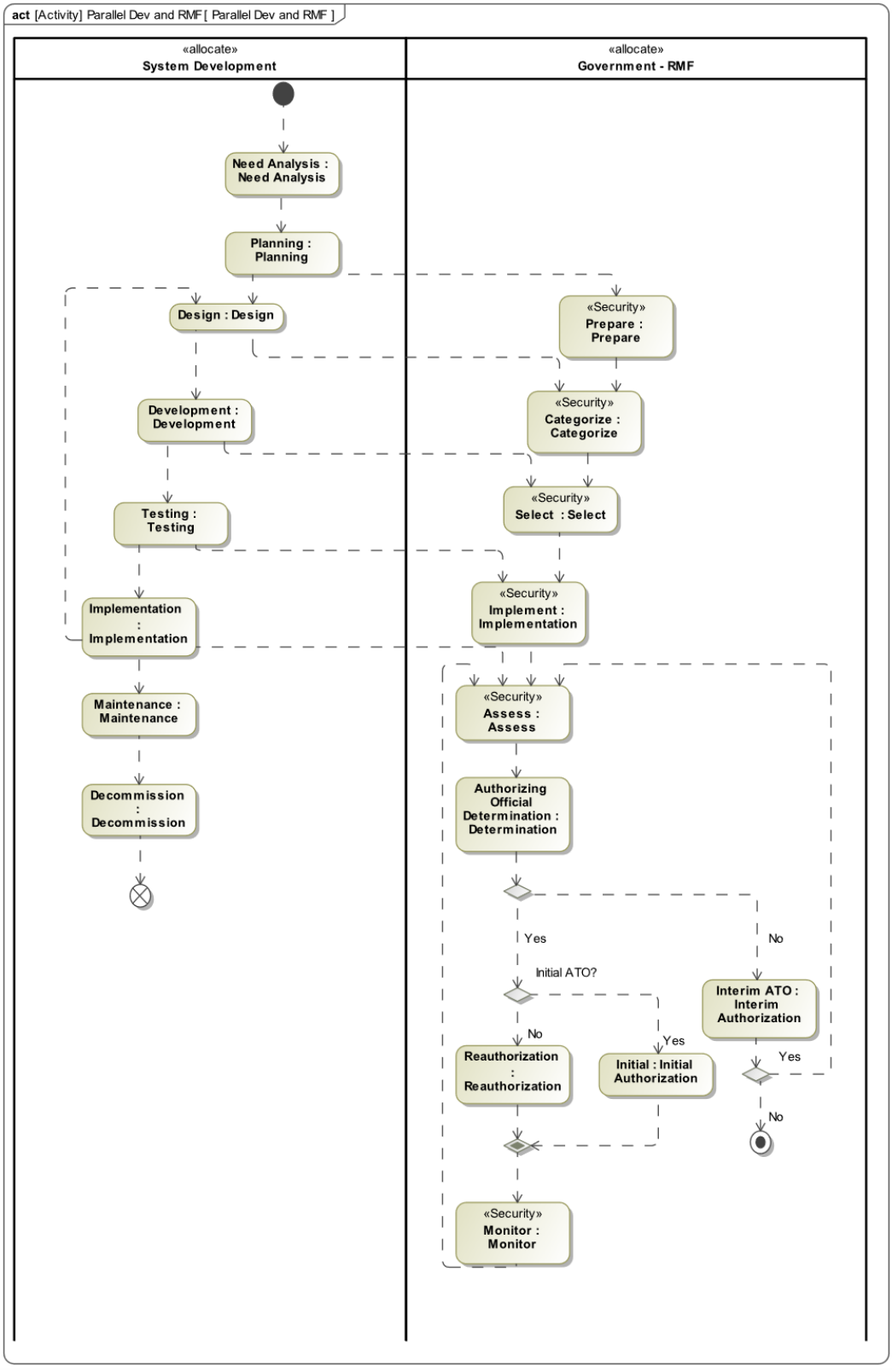



*Figure 43 - Activity Diagram - Parallel Development and RMF*

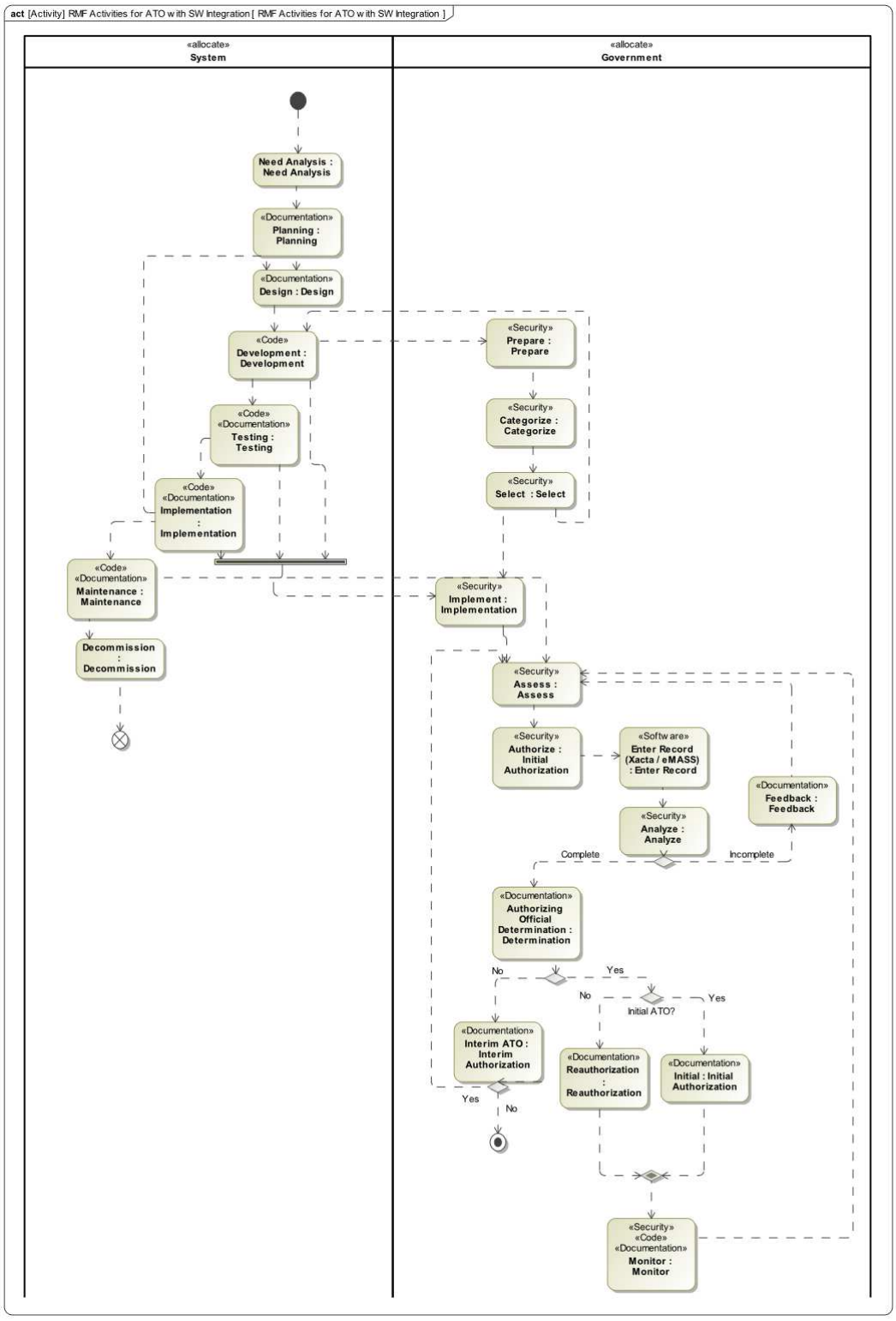



*Figure 44 - Activity Diagram – RMF Activities for ATO with Software Integration*

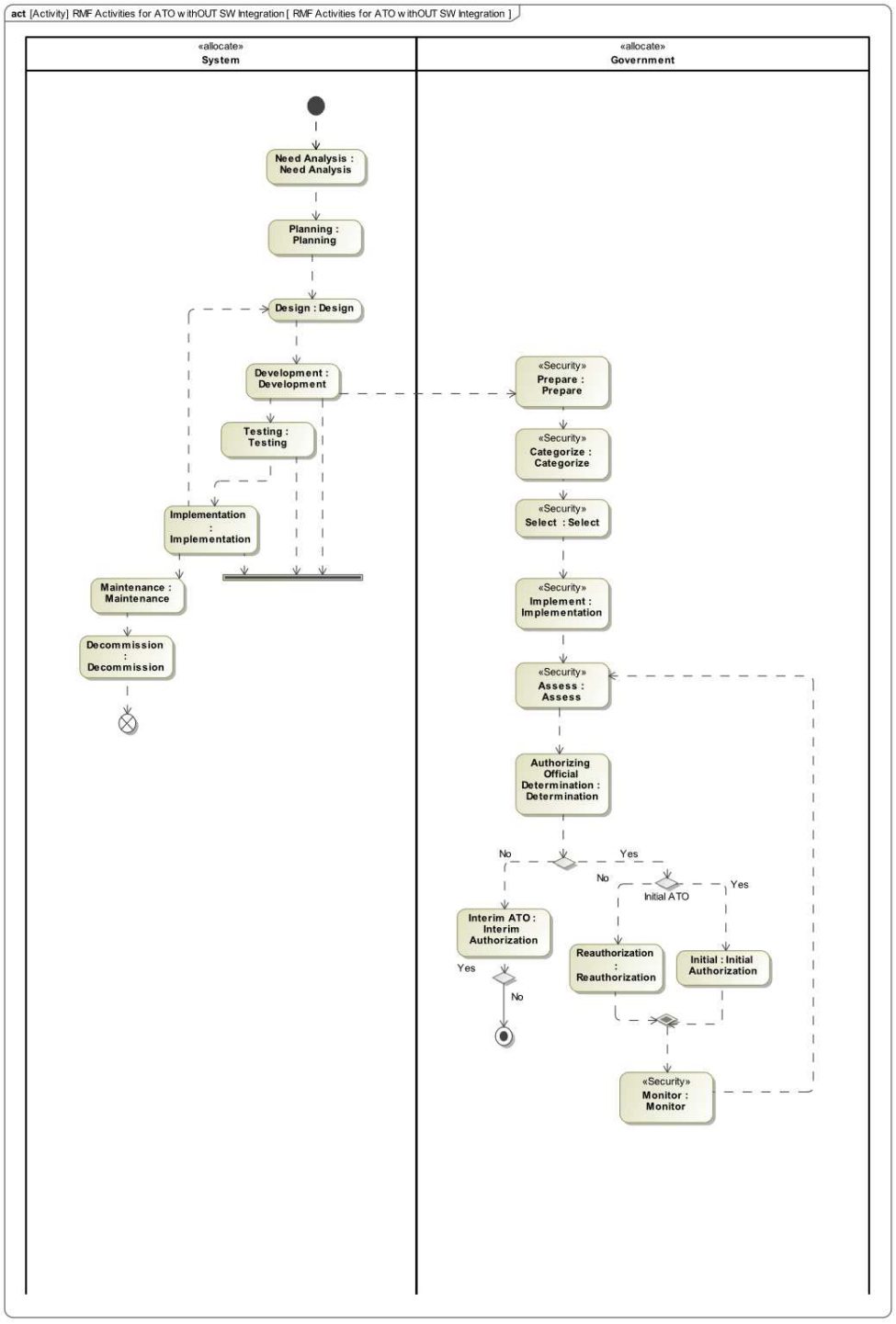



*Figure 45 - Activity Diagram - RMF Activities for ATO without Software Integration*

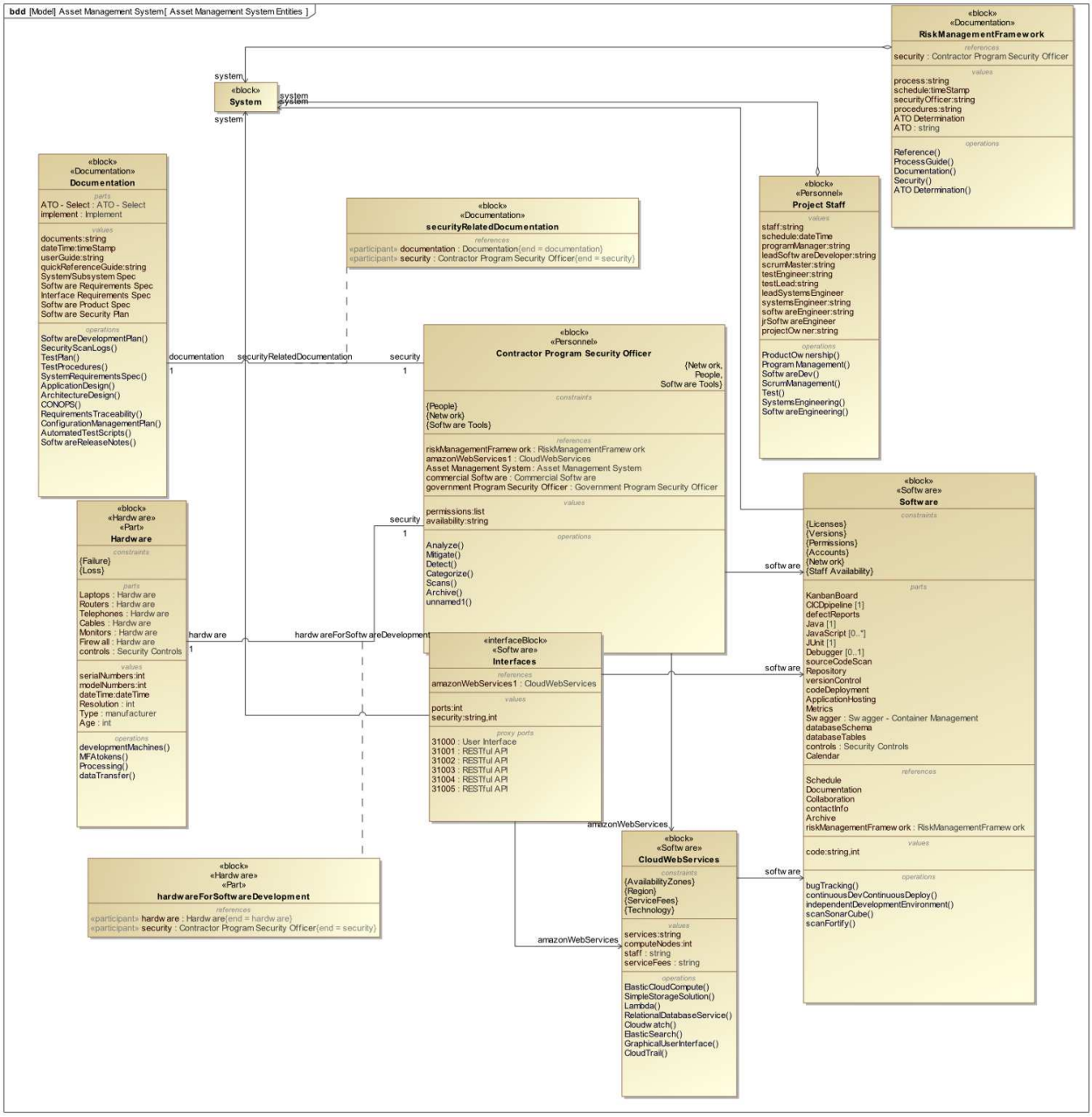



*Figure 46 - Block Definition Diagram – Asset Management System Entities*

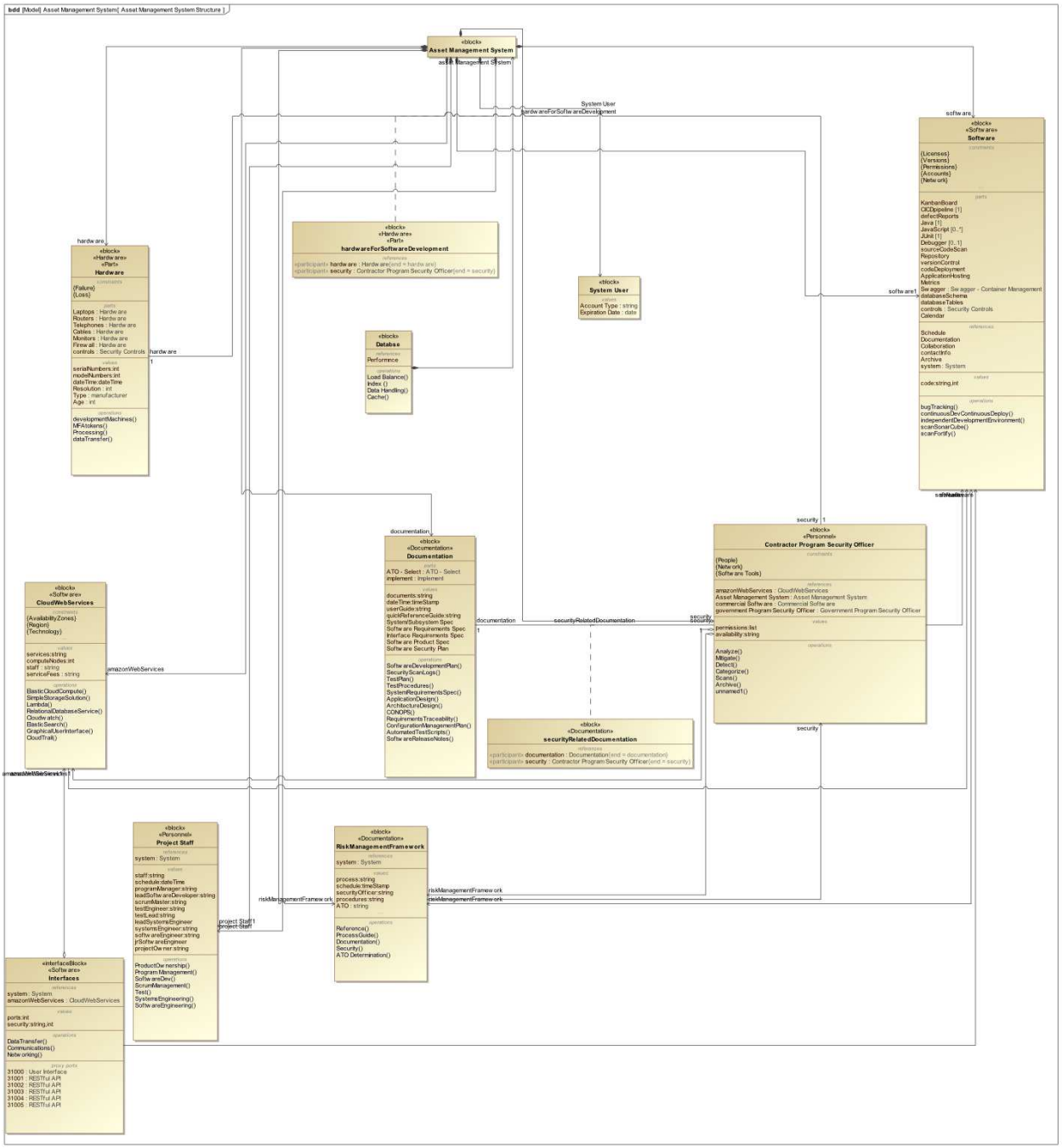



*Figure 47 - Block Definition Diagram - Asset Management System Structure*

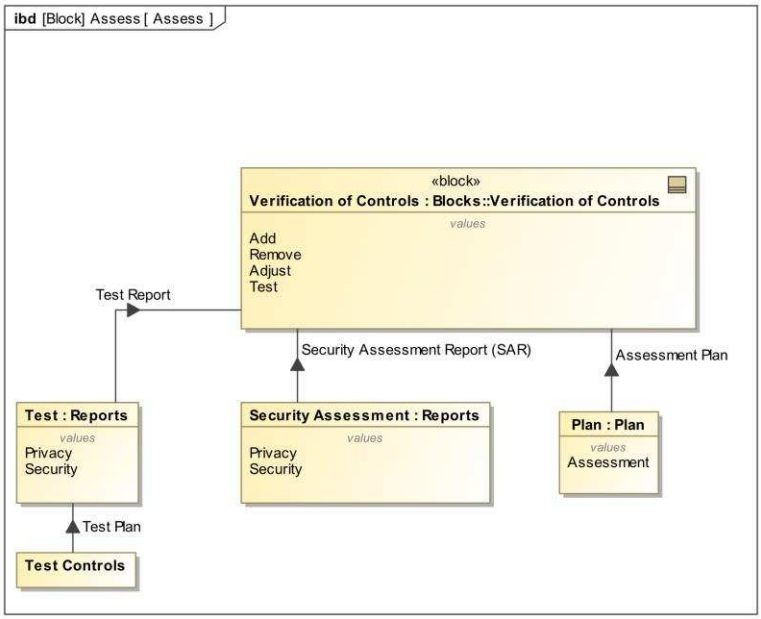



*Figure 48 - ATO Assess Step Structure Diagram*

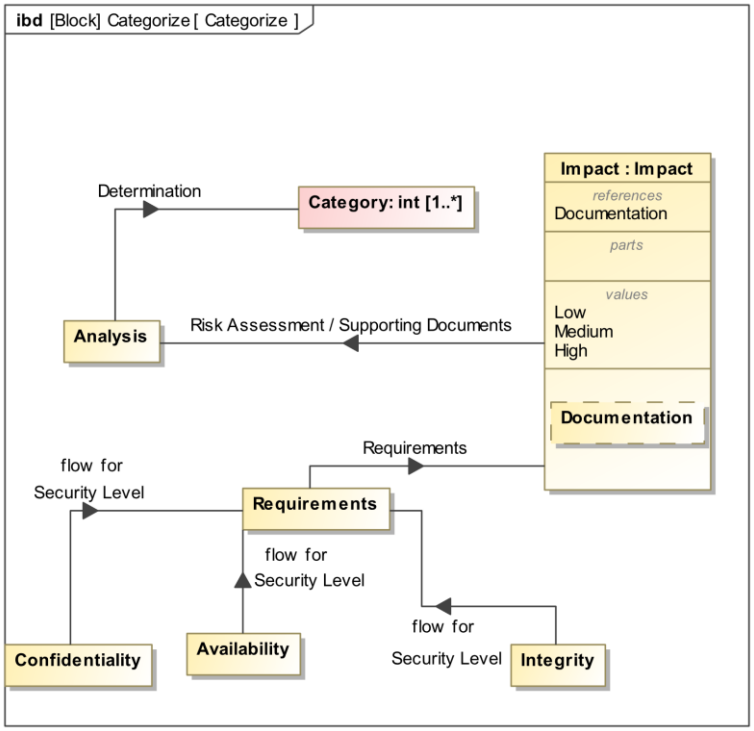



*Figure 49 - ATO Categorize Step Structure Diagram*

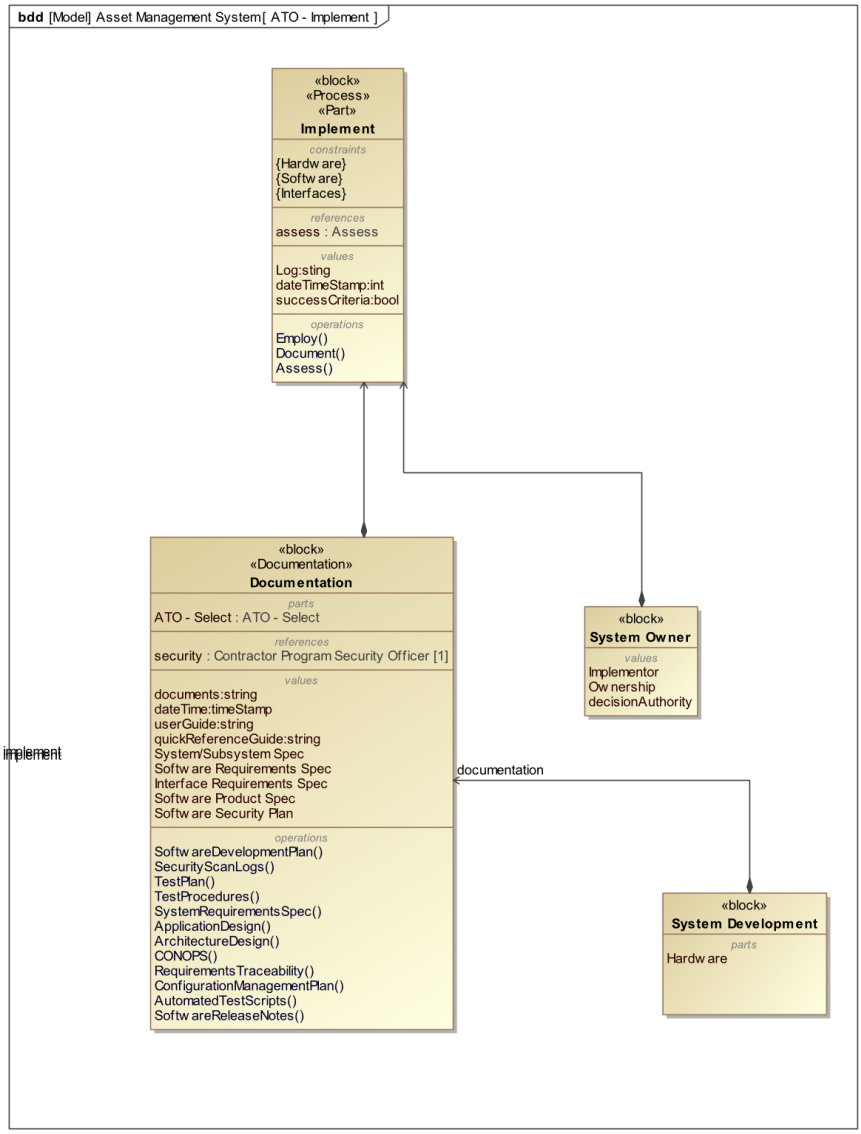



*Figure 50 - ATO Implement Step Structure Diagram*

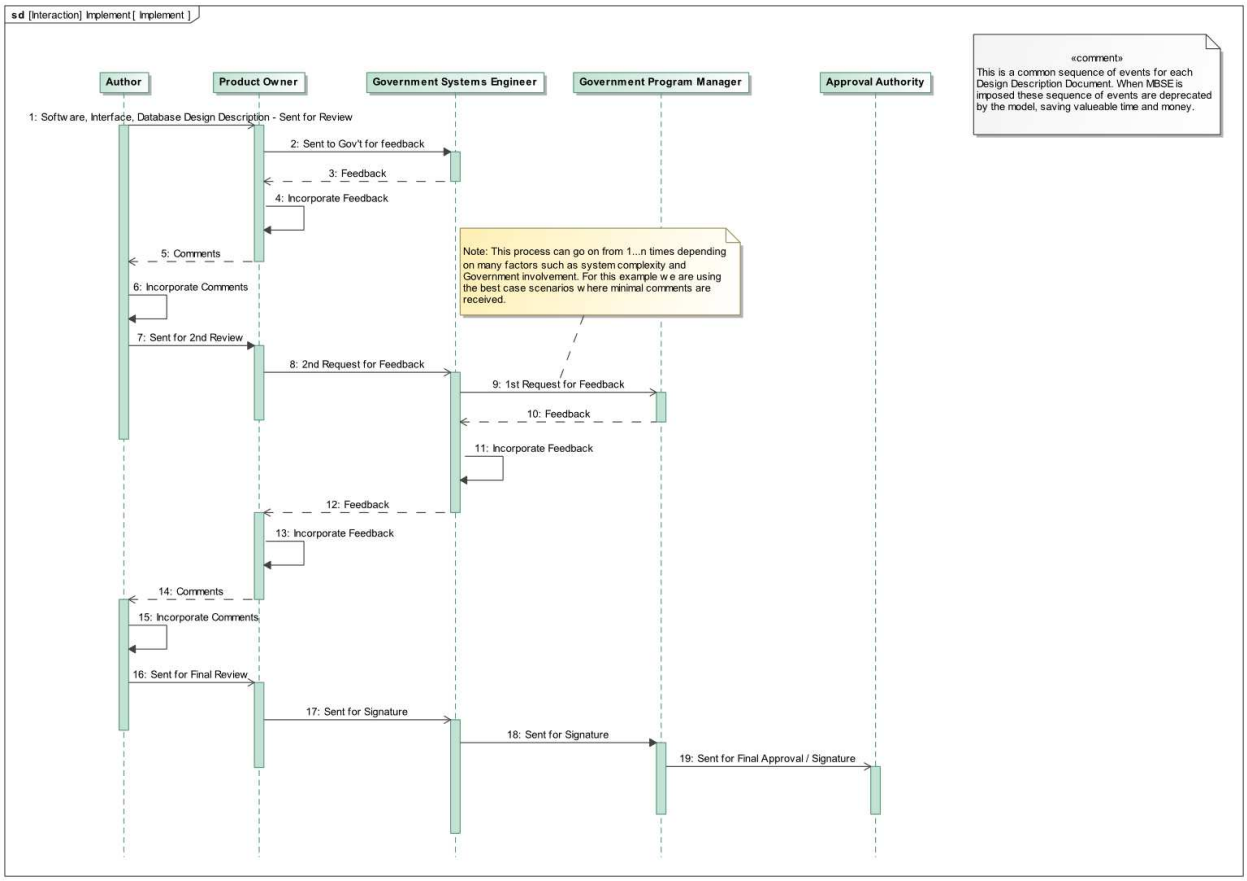



*Figure 51 - Implement Step - Software Interface Database Design Description Document*

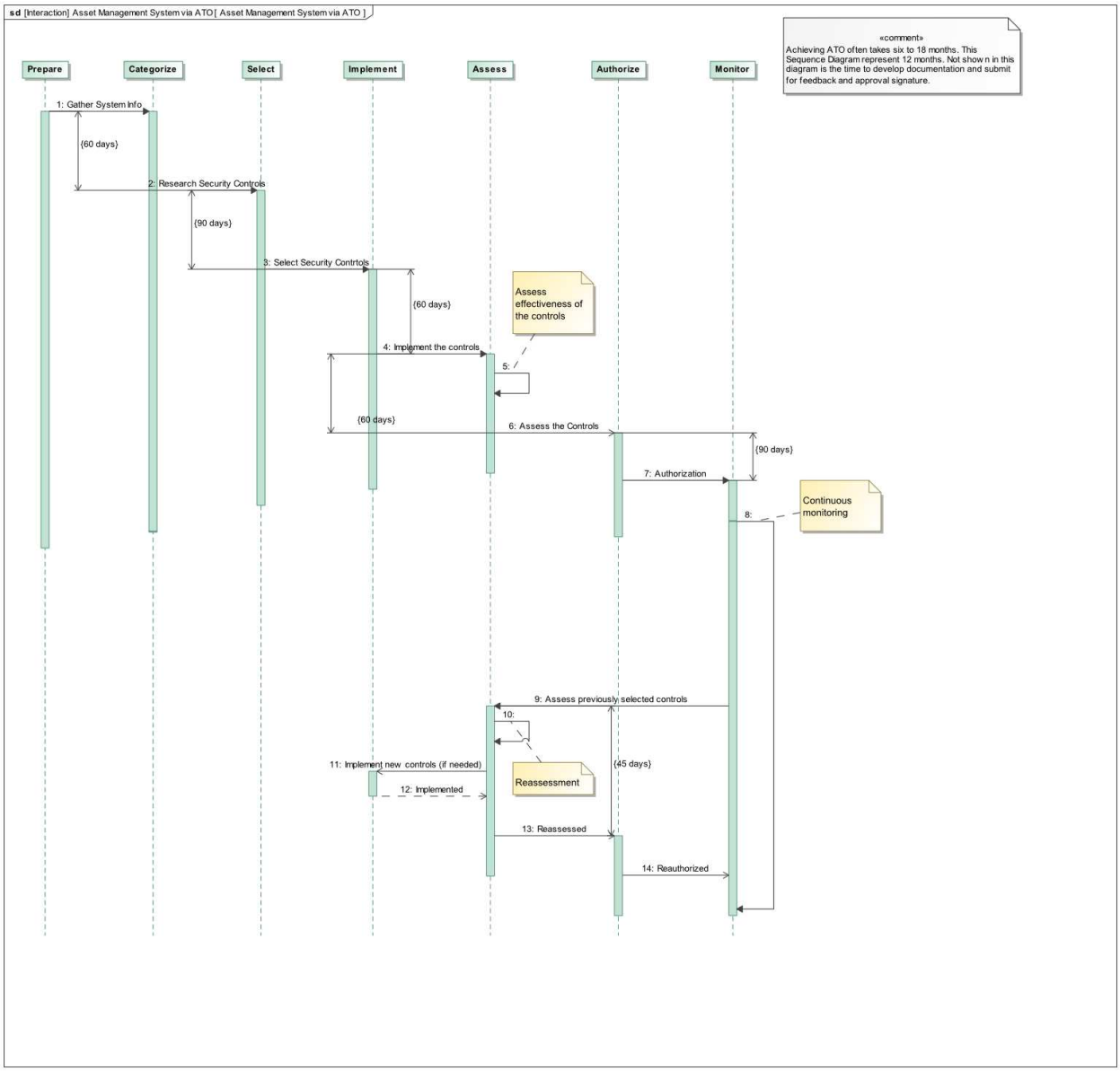



*Figure 52 - Asset Management ATO Timing Analysis*

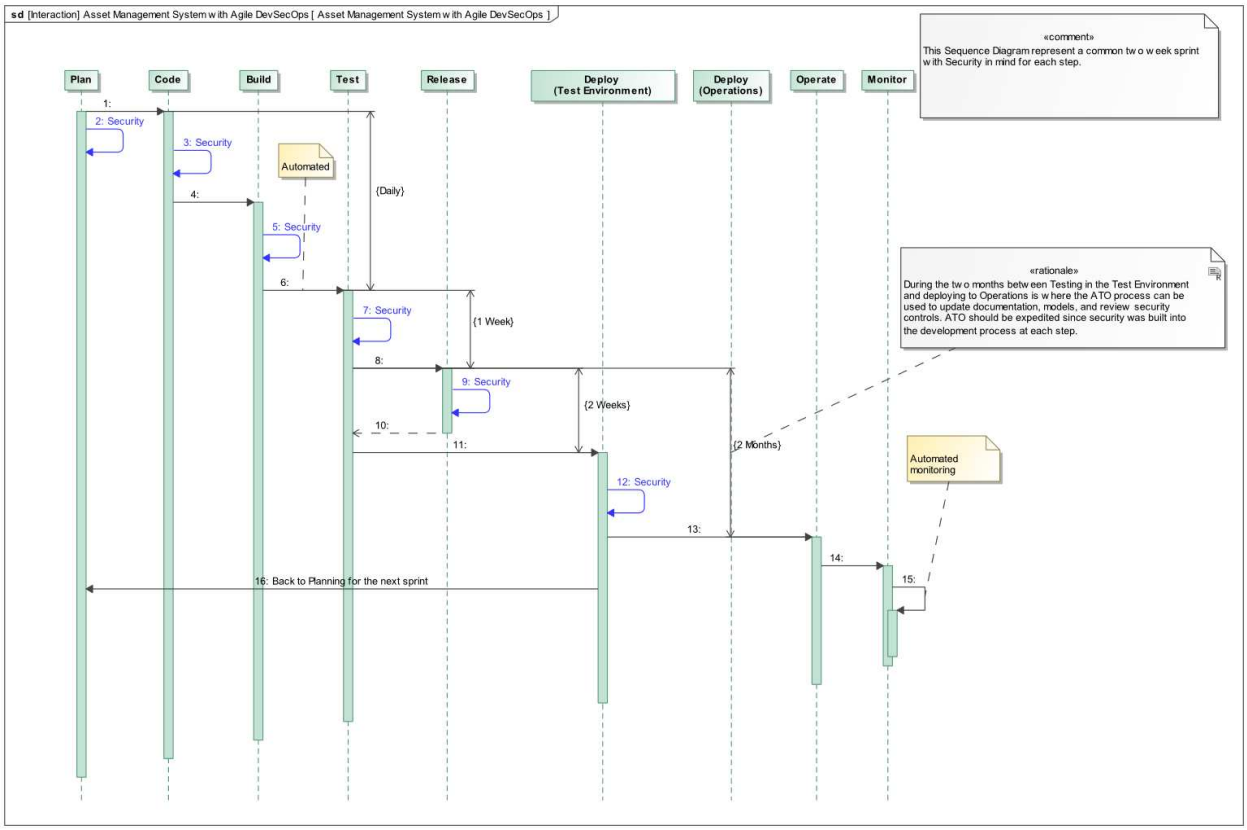



*Figure 53 - Asset Management System with DevSecOps & ATO Processes Sequence Diagram Example*

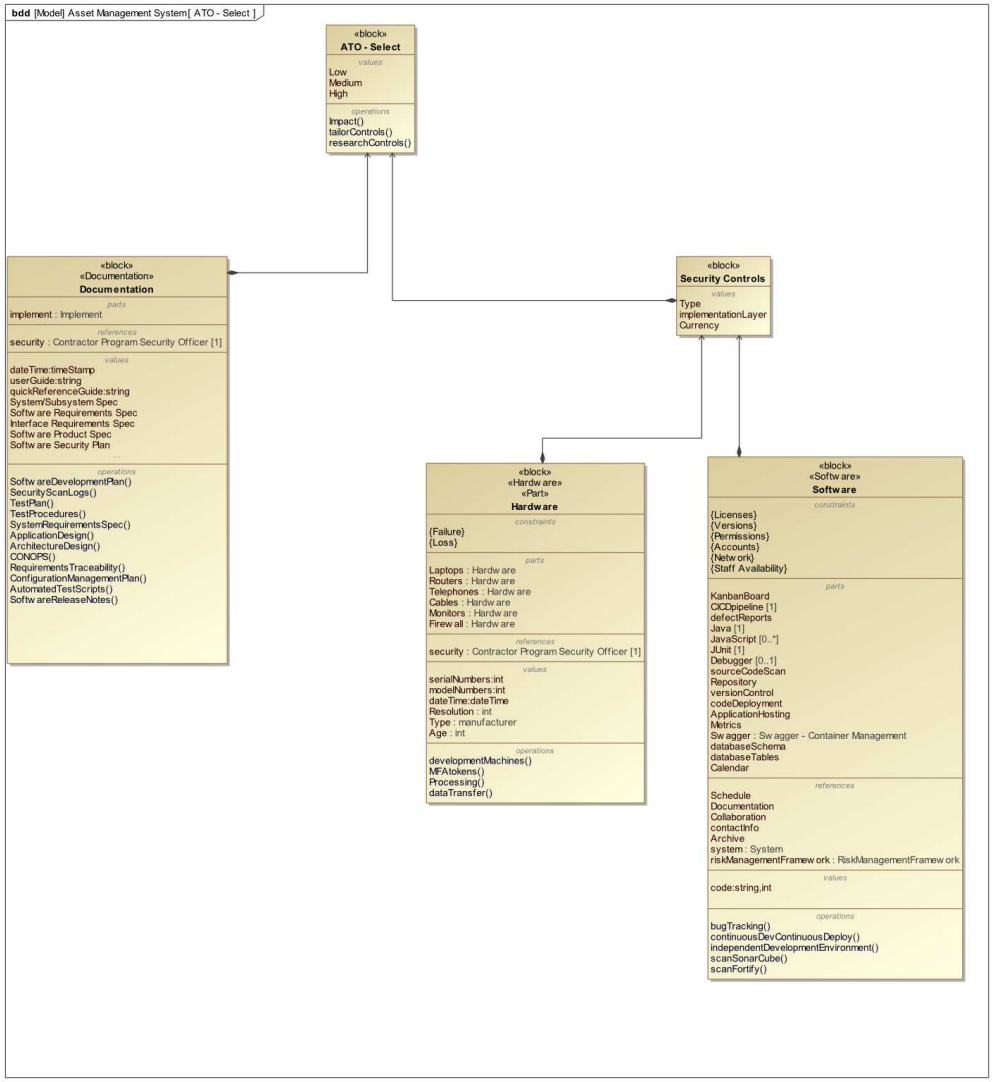



*Figure 54 - ATO Select Step Block Definition Diagram*

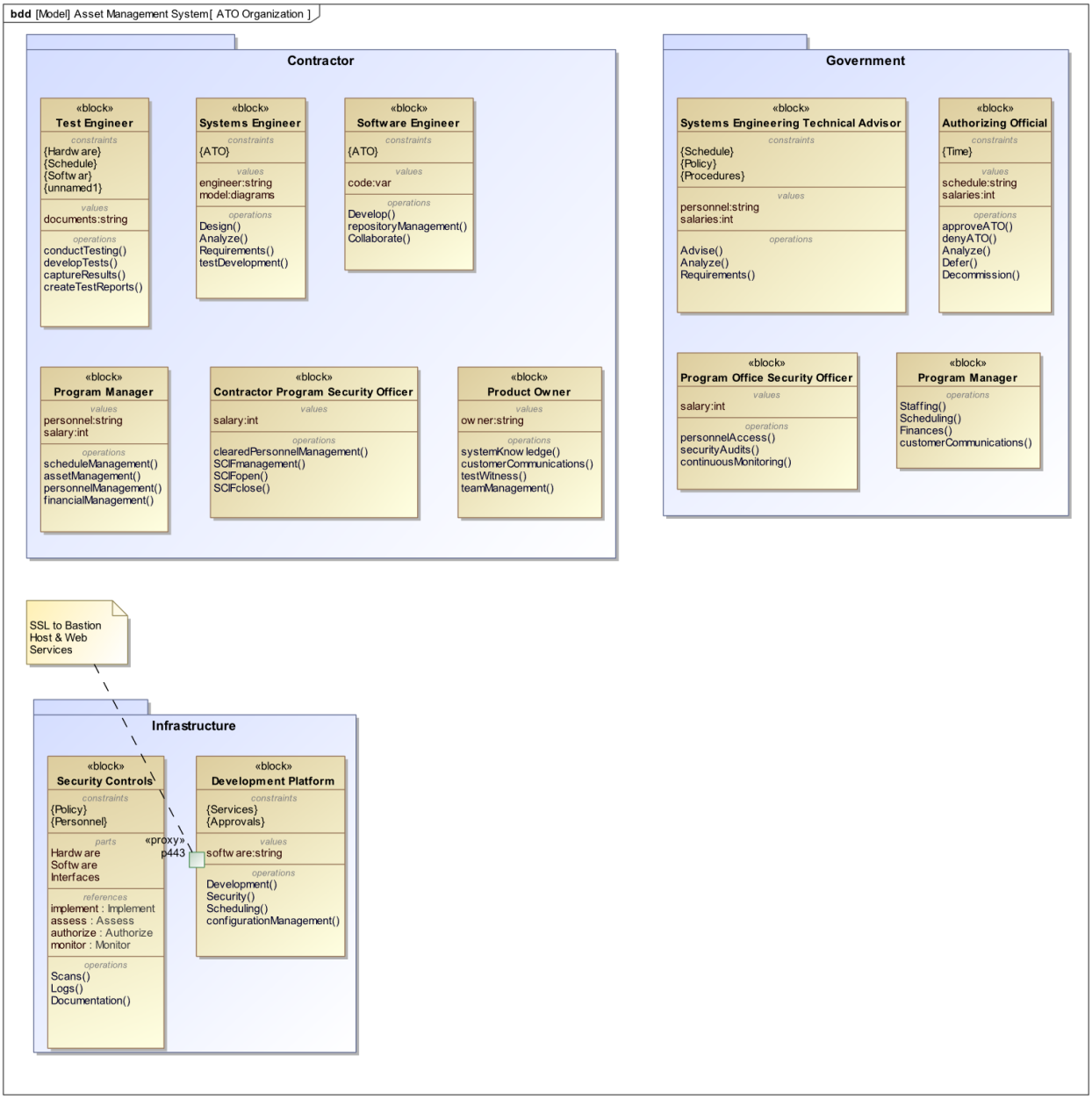
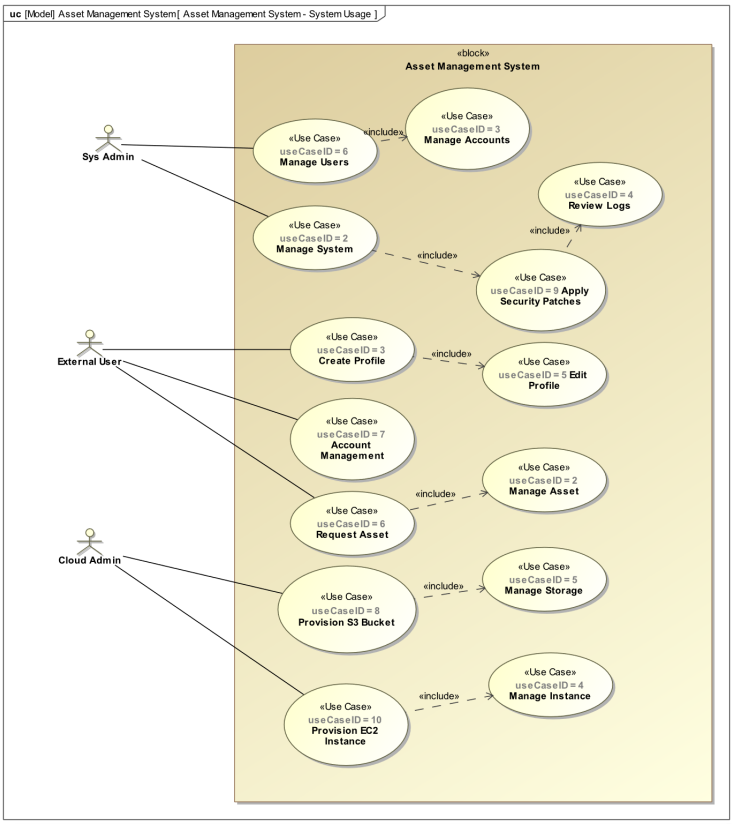



*Figure 55 - Asset Management System and ATO Structure*



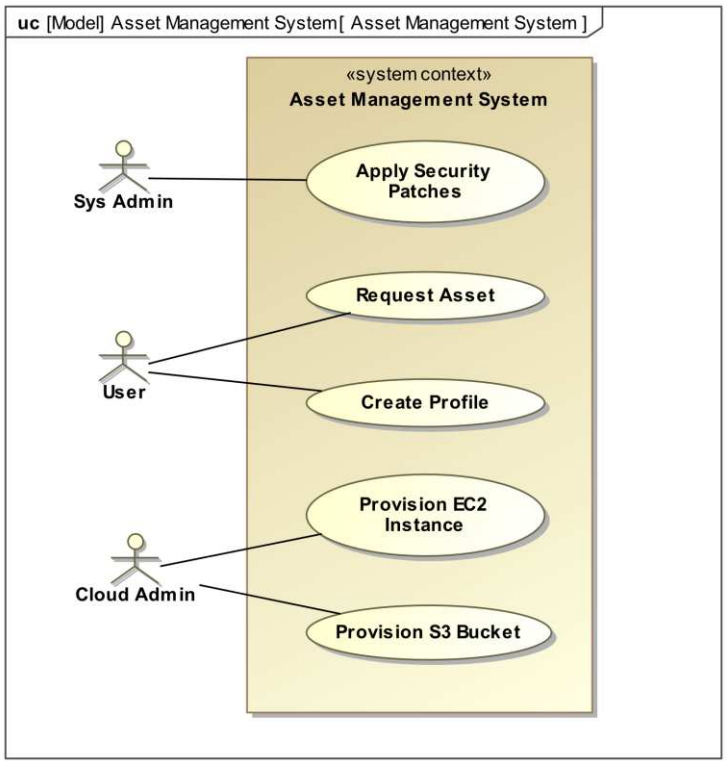




*Figure 56 - Asset Management System Usage*

# APPENDIX C - ACRONYMS

| | |
|---|---|
| AC | Access Control |
| AI | Artificial Intelligence |
| AO | Authorizing Official |
| AT | Awareness and Training |
| ATO | Authorization to Operate |
| AU | Audit and Accountability |
| AWS | Amazon Web Services |
| BDD | Block Definition Diagram |
| CA | Security Assessment and Authorization |
| C&A | Certification and Accreditation |
| CIA | Confidentiality, Integrity, Availability |
| CISSP | Certified Information System Security Professional |
| CM | Configuration Management |
| CNSSI | Committee on National Security Systems Instruction |
| CONOPS | Concept of Operations |
| COTS | Commercial off the Shelf |
| CP | Contingency Planning |
| CR | Change Request |
| CSV | Comma Separated Value |
| DAA | Designated Accrediting Authority |
| DB | Database |
| DBSE | Document Based Systems Engineering |
| DE | Digitals Engineering |
| DIACAP | Dept of Defense Information Assurance Certification and Accreditation Process |
| DII | Defense Information Infrastructure |
| DITSCAP | Dept of Defense Information Technology Security Certification and Accreditation Process |
| DOD | Department of Defense |
| FAQ | Frequently asked Question |
| FFRDC | Federally Funded Research and Development Center |
| GOTS | Government of the Shelf |
| IA | Information Assurance |
| IBD | Internal Block Diagram |
| ICD | Interface Control Document |

| | |
|---|---|
| ID | Identification |
| IEEE | Institute of Electrical and Electronics Engineers |
| INCOSE | International Council of Systems Engineers |
| IDE | Independent Development Environment |
| INL | Idaho National Labs |
| IR | Incident Response |
| IS | Information System |
| ISCM | Information Security Continuous Monitoring |
| IT | Information Technology |
| KPP | Key Performance Parameters |
| MA | Maintenance |
| MBD | Model-Based Development |
| MBSE | Model-Based Systems Engineering |
| MBSR | Model-Based Structured Requirements |
| MILCOM | Military Command |
| MP | Media Protection |
| NIST | National Institute of Standards and Technology |
| OMG | Object Management Group |
| OOD | Object-Oriented Design |
| OOP | Object-Oriented Programming |
| PAA | Principal Accrediting Authority |
| PE | Physical and Environmental Protection |
| PII | Personally Identifiable Information |
| PL | Planning |
| PM | Program Manager |
| PS | Personnel Security |
| QA | Quality Assurance |
| RA | Risk Assessment |
| RMF | Risk Management Framework |
| RQ | Research Question |
| RTM | Requirements Traceability Matrix |
| SA | System and Services Accreditation |
| SC | System and Communication Protection |
| SDR | System Design Review |
| SE | Systems Engineering |
| SELC | Systems Engineering Life Cycle |
| SI | Systems Integration |
| SLA | Service Level Agreement |

| | |
|---|---|
| SME | Subject Matter Expert |
| SOA | Service Oriented Architecture |
| SP | Special Publication |
| STD | Software Test Description |
| STE | Staff Years of Technical Effort |
| SYS | Systems |
| TO | Task Order |
| UML | Unified Modeling Language |
| USAISEC | US Army Information Systems Engineering Command |
| USG | United States Government |
| VTL | Velocity Template Language |

# APPENDIX D – ASSET MANAGEMENT SYSTEM EXAMPLE REQUIREMENTS

The table below provides high-level requirements used to model the Asset Management System. These are representative of requirements that can be further decomposed to derived requirements which are interpretations or inferred.

*Table 10 - Comma Separated Value Example Requirements*

| Req ID | Type | Description |
|---|---|---|
| **1.0** | **Functional** | |
| 1.1 | Functional | User Authentication and Authorization |
| 1.2 | Functional | Asset Registration, Update, and Deletion |
| 1.3 | Functional | Asset Search and Filtering |
| 1.4 | Functional | Reporting and Analytics |
| 1.5 | Functional | Role-based Access Control |
| 1.6 | Functional | Integration with External Systems (e.g., ERP, CRM) |
| 1.7 | Functional | Mobile Accessibility |
| 1.8 | Functional | Notification System (Email, SMS, Push Notifications) |
| 1.9 | Functional | Data Export and Import (CSV, Excel) |
| 1.10 | Functional | Audit Trail and Logging |
| 1.11 | Functional | Dashboard with Key Metrics |
| 1.12 | Functional | User Management (Create, Update, Delete Users) |
| **2.0** | **Non-Functional** | |
| 2.1 | Non-Functional | High Availability and Reliability |
| 2.2 | Non-Functional | Scalability to Handle Growing Number of Assets and Users |
| 2.3 | Non-Functional | User-Friendly Interface |
| 2.4 | Non-Functional | Cross-Browser Compatibility |
| 2.5 | Non-Functional | Multi-Language Support |
| 2.6 | Non-Functional | Compliance with Industry Standards (e.g., ISO, GDPR) |
| 2.7 | Non-Functional | Regular Software Updates and Maintenance |
| 2.8 | Non-Functional | Backup and Disaster Recovery |
| 2.9 | Non-Functional | Customizable Interface and Workflows |

| | | |
|---|---|---|
| 2.10 | Non-Functional | Low Latency in User Interactions |
| **3.0** | **Performance** | |
| 3.1 | Performance | System Should Support Concurrent Users without Performance Degradation |
| 3.2 | Performance | Response Time for Search Queries Should be < 2 Seconds |
| 3.3 | Performance | Data Load Time Should be < 3 Seconds |
| 3.4 | Performance | System Uptime Should be 99.9% or Higher |
| 3.5 | Performance | Support for Bulk Data Operations |
| 3.6 | Performance | Efficient Resource Utilization |
| 3.7 | Performance | Performance Metrics and Monitoring |
| **4.0** | **Security** | |
| 4.1 | Security | Data Encryption in Transit and at Rest |
| 4.2 | Security | Regular Security Audits and Vulnerability Assessments |
| 4.3 | Security | Role-Based Access Control and Permissions Management |
| 4.4 | Security | Multi-Factor Authentication (MFA) |
| 4.5 | Security | Compliance with Security Standards (e.g., SOC 2, ISO 27001) |
| 4.6 | Security | Secure API Endpoints |
| 4.7 | Security | Incident Response Plan |
| 4.8 | Security | User Activity Monitoring and Alerts |
| 4.9 | Security | Data Loss Prevention (DLP) |
| 4.1 | Security | Secure Backup and Recovery Processes |
| **5.0** | **Personnel** | |
| 5.1 | Personnel | Project Manager |
| 5.2 | Personnel | Full Stack Developers |
| 5.3 | Personnel | UX/UI Designers |
| 5.4 | Personnel | QA/Test Engineers |
| 5.5 | Personnel | Security Specialists |
| 5.6 | Personnel | DevOps Engineers |
| 5.7 | Personnel | Database Administrators |
| 5.8 | Personnel | System Administrators |
| 5.9 | Personnel | Technical Support Staff |
| 5.10 | Personnel | Business Analysts |
| 5.11 | Personnel | Compliance Officers |

*Table 11 - Cameo Systems Modeler Example Requirements Table*

| # | Docum... | Name | Text | Id | Verified By |
|---|---|---|---|---|---|
| 1 | | 1.0 Functional Requirements | Asset Management System Functional Requirements | 1.0 | Functional Requirements Test Procedure |
| 2 | | 1.5 | The system shall track asset maintenance schedules and send reminders to appropriate users | 1.5 | Functional Requirements Test Procedure<br>Reports and Logging |
| 3 | | 1.1 | The system shall allow users to create and manage asset records | 1.1 | Functional Requirements Test Procedure<br>Record Management() |
| 4 | | 1.2 | The system shall provide search functionality for users to easily locate specific assets | 1.2 | Functional Requirements Test Procedure<br>Record Search()<br>User Interface |
| 5 | | 1.3 | The system shall generate reports on asset inventory, usage, and depreciation | 1.3 | Functional Requirements Test Procedure<br>Reports and Logging |
| 6 | | 1.4 | The system shall support multiple user roles with varying levels of access permissions | 1.4 | Functional Requirements Test Procedure<br>3.1 Security<br>User Permission() |
| 7 | | 1.6 | The system shall allow for bulk importing and exporting of asset data | 1.6 | Functional Requirements Test Procedure<br>Data Import & Export |
| 8 | | 1.7 | The system shall integrate with existing third-party systems for data exchange | 1.7 | Functional Requirements Test Procedure<br>Data Sharing() |
| 9 | | 1.8 | The system shall provide a dashboard with key performance indicators for asset management | 1.8 | Functional Requirements Test Procedure<br>User Interface |
| 10 | | 1.9 | The system shall allow for customization of asset categories and attributes | 1.9 | Functional Requirements Test Procedure<br>Interface Perferences() |
| 11 | | 1.10 | The system shall automatically assign unique asset identifiers to newly added assets | 1.10 | Functional Requirements Test Procedure<br>Data ID() |
| 12 | | 2.0 Non-Functional Requirements | Asset Management System Non-Functional Requirements | 2.0 | Non-Functional Requirements Test Procedure |
| 13 | | 2.1 | The system shall be accessible from any device with internet connectivity | 2.1 | Non-Functional Requirements Test Procedure<br>Database<br>User Interface |
| 14 | | 2.2 | The system shall have a responsive user interface that adapts to different screen sizes | 2.2 | Non-Functional Requirements Test Procedure<br>User Interface<br>508 Compliance() |
| 15 | | 2.3 | The system shall have a documented user manual and training materials available for users | 2.3 | Non-Functional Requirements Test Procedure<br>Documentation |
| 16 | | 2.4 | The system shall have a backup and disaster recovery plan in place | 2.4 | Non-Functional Requirements Test Procedure<br>Documentation<br>Disaster Recovery() |
| 17 | | 2.5 | The system shall comply with relevant data protection and privacy regulations | 2.5 | Non-Functional Requirements Test Procedure<br>Documentation |
| 18 | | 2.6 | The system shall have a service level agreement guaranteeing uptime and performance | 2.6 | Non-Functional Requirements Test Procedure<br>Documentation |
| 19 | | 2.7 | The system shall have a scalable architecture to accommodate growing data and user needs | 2.7 | Non-Functional Requirements Test Procedure<br>Documentation |
| 20 | | 2.8 | The system shall have a logging and auditing mechanism to track user actions and system events | 2.8 | Non-Functional Requirements Test Procedure<br>Reports and Logging |
| 21 | | 2.9 | The system shall provide a mechanism for users to provide feedback and request new features | 2.9 | Non-Functional Requirements Test Procedure<br>User Interface<br>Documentation |
| 22 | | 3.0 Security Requirements | Asset Management System Security Requirements | 3.0 | Security Test Procedure |
| 23 | | 3.1 | The system shall encrypt all data at rest and in transit | 3.1 | Verify the Data is Encrpyted<br>Data Encryption()<br>Security Test Procedure |
| 24 | | 3.2 | The system shall implement role-based access control to restrict unauthorized access to sensitive data | 3.2 | Check User Permissions()<br>User Lookup()<br>Security Test Procedure |
| 25 | | 3.3 | The system shall provide secure user authentication mechanisms, such as multi-factor authentication | 3.3 | Check Multifactor Authentication()<br>Security Test Procedure |
| 26 | | 3.4 | The system shall monitor for and alert on any unauthorized access attempts or suspicious activity | 3.4 | Security Audits<br>Security Test Procedure |
| 27 | | 3.5 | The system shall regularly undergo security assessments and vulnerability scans | 3.5 | Security Audits<br>Security Test Procedure |
| 28 | | 3.6 | The system shall have mechanisms in place to prevent data breaches and protect against malicious attacks | 3.6 | Database Security<br>Security Test Procedure |
| 29 | | 3.7 | The system shall enforce strong password policies for user accounts. Ref; User, System, Password | 3.7 | User Authentication<br>Security Test Procedure |
| 30 | | 3.8 | The system shall have a process for securely disposing of assets and their associated data | 3.8 | Verify Data Disposal()<br>Security Test Procedure |
| 31 | | 3.9 | The system shall restrict access to sensitive data based on user roles and permissions | 3.9 | Check User Permissions()<br>Security Test Procedure |

| | | | | | |
|---|---|---|---|---|---|
| 32 | | R 4.0 Perfomance Requirements | Asset Management System Performance Requirements | 4.0 | Performance Test Procedure |
| 33 | | P 4.1 | The system shall be capable of handling a large number of concurrent users without experiencing performance degradation | 4.1 | Performance Test Procedure<br>User Permission()<br>Load Balancing() |
| 34 | | P 4.2 | The system shall have efficient database queries and data retrieval processes for quick response times | 4.2 | Performance Test Procedure<br>Data Handling() |
| 35 | | P 4.3 | The system shall be able to handle large amounts of data without impacting performance | 4.3 | Performance Test Procedure<br>Database |
| 36 | | P 4.4 | The system shall have automated processes for optimizing system performance, such as database indexing | 4.4 | Performance Test Procedure<br>Index() |
| 37 | | P 4.5 | The system shall have mechanisms in place to monitor and maintain optimal system performance | 4.5 | Performance Test Procedure<br>Load Balancing()<br>Reports and Logging |
| 38 | | P 4.6 | The system shall have a mechanism for caching frequently accessed data to improve performance | 4.6 | Performance Test Procedure<br>Cache() |
| 39 | | P 4.7 | The system shall have mechanisms for load balancing and distributing workload evenly across servers | 4.7 | Performance Test Procedure<br>Load Balancing() |
| 40 | | P 4.8 | The Asset Management System shall maintain 99.9% availability under normal operating conditions. | 4.8 | |
| 41 | | R 5.0 Personnel Requirements | Asset Management System Personnel Requirements | 5.0 | Personnel Test Procedure |
| 42 | | R 5.1 | The system shall have a designated system administrator responsible for managing user accounts and system configuration | 5.1 | Personnel Test Procedure<br>Accounts |
| 43 | | R 5.2 | The system shall have a help desk or support team available to assist users with technical issues | 5.2 | Personnel Test Procedure<br>Help Desk |
| 44 | | R 5.3 | The system shall have trained personnel responsible for conducting regular data backups and disaster recovery procedures | 5.3 | Personnel Test Procedure<br>Training |
| 45 | | R 5.4 | The system shall have personnel responsible for monitoring system performance and addressing any performance issues that arise | 5.4 | Personnel Test Procedure |
| 46 | | R 5.5 | The system shall have a dedicated security team responsible for monitoring and mitigating security threats | 5.5 | Personnel Test Procedure<br>Security Monitoring() |
| 47 | | R 5.6 | The system shall have trainers available to conduct user training sessions and provide support as needed | 5.6 | Personnel Test Procedure<br>Training Material() |
| 48 | | R 5.7 | The system shall have personnel responsible for conducting regular security assessments and implementing necessary security measures | 5.7 | Personnel Test Procedure<br>Threat Mitigation() |
| 49 | | R 5.8 | The system shall have personnel responsible for maintaining system documentation and ensuring it is up to date and accurate | 5.8 | Personnel Test Procedure<br>Documentation Management() |
| 50 | | R 5.9 | The system shall have personnel available to implement new features and enhancements based on user feedback and requirements | 5.9 | Personnel Test Procedure<br>Software Development() |