A Cost Model for Testing Unmanned and Autonomous Systems of Systems

By

Indira D. Deonandan

Bachelor of Science in Engineering Science
Smith College, 2008

Submitted to:
The Department of Aeronautics and Astronautics and
The Engineering Systems Division
in Partial Fulfillment of the Requirements for the Dual Degree in

Master of Science in Aeronautics and Astronautics
and
Master of Science in Technology and Policy

at the
Massachusetts Institute of Technology
February 2011

Signature of Author ____________________
Department of Aeronautics and Astronautics and Engineering Systems Division
22 January, 2011

Certified by ____________________
Ricardo Valerdi
Research Associate, Engineering Systems Division
Thesis Supervisor

Certified by ____________________
Deborah Nightingale
Professor of the Practice, Aeronautics and Astronautics and Engineering Systems
Thesis Reader

Accepted by ____________________
Eytan Modiano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Accepted by ____________________
Dava Newman
Professor of Aeronautics and Astronautics and Engineering Systems
Director, Technology and Policy Program

(THIS PAGE INTENTIALLY LEFT BLANK)

A Cost Model for Testing Unmanned and Autonomous Systems of Systems

Indira D. Deonandan

Submitted to:
The Department of Aeronautics and Astronautics and
The Engineering Systems Division
in Partial Fulfillment of the Requirements for the Dual Degree in

Master of Science in Aeronautics and Astronautics
and
Master of Science in Technology and Policy

Abstract

The evolutionary nature of Unmanned and Autonomous Systems of Systems (UASoS) acquisition needs to be matched by equally evolutionary test capabilities in the future. There is currently no standard method to determine what is required to make programs safe for deployment, nor is there the ability to make effective contingency plans should testing requirements change. Spending too much effort designing goals when causal understandings are still in flux is inefficient. As such, policy making and enforcing policies on the deployment of UASoS becomes very problematic.

Testing is required especially for UASoS to identify risk, improve capabilities and minimize unpleasant surprises. It needs to be effective and focused, determining the issues and working towards ensuring the risks of the UASoS are known. It is important to have adequate feedback loops, a culture of information sharing and learning from best practices, as well as the development of metrics and/or performance indicators that adequately reflect the effectiveness of the test process.

This thesis describes a model that is part of a larger Prescriptive and Adaptive Testing Framework (PATFrame), which uses knowledge acquisition to minimize risk through a decision support system. This work presents the cost and risk considerations for UASoS T&E and provides the preliminary parameters to conduct trade-off analyses for T&E. It also provides guidance on how the DoD can adopt such tools to transform the DoD T&E enterprise. The model is a combination of information collected from various normative and descriptive views of testing based on literature review, surveys, and interviews with members of the Department of Defense (DoD) T&E community

A cost estimation model can have significant impacts on how the DoD currently does testing and would help maximize the use of the resources available. It is a model based method for calculating effort for test and evaluation and forms a baseline for strategic decision making in DoD acquisition programs. The intent is to predict within a certain probability that a test program can be completed within a certain budget given the assumptions used in characterizing the UASoS and the T&E process.

Thesis Supervisor: Ricardo Valerdi
Title: Research Associate, Engineering Systems Division

Thesis Reader: Deborah J. Nightingale
Title: Professor of the Practice, Aeronautics and Astronautics and Engineering Systems

(THIS PAGE INTENTIONALLY LEFT BLANK)

Dedication

This thesis is dedicated to my family,
My father – Vishnudatt, My mother – Serajie, and My sisters – Savitri and Chandra

Table of Contents

List of Figures

List of Tables

Abbreviations

ATEC	Army Test and Evaluation Center
AFOTEC	Air Force Operational Test and Evaluation Center
CER	Cost Estimation Relationship
COCOMO II	Constructive Cost Model version II
COCOTS	Constructive Commercial-off-the-shelf Model
COI	Critical Operational Issue
COSOSIMO	Constructive System-of-systems Cost Model
COSYSMO	Constructive Systems Engineering Cost Model
COTS	Commercial Off The Shelf
DoD	Department of Defense
DT	Developmental Testing
EM	Effort Multiplier
EMR	Effort Multiplier Ratio
GAO	Government Accountability Office
KPP	Key Performance Parameter
OT	Operational Testing
OTC	Operational Test Command
PATFrame	Prescriptive and Adaptive Testing Framework
PM	Person Month
S&T	Science and Technology
SoS	System of System
T&E	Test and Evaluation
TRMC	Test Resource Management Center
UAS	Unmanned and Autonomous System
UASoS	Unmanned and Autonomous System of System

Biographical Note

Indira Deonandan graduated *Cum laude* from Smith College in 2008, with a Bachelors of Science in Engineering Science with a concentration in Mechanical and Aerospace Engineering from Princeton University.

Publications:

First Author

1. October 2010, *"The Impact of Professional Societies on the Advancement of the Profession"*, Working Journal Paper for the Journal of Management Studies on Professions and Institutional Change
2. July 2010, *"The Impact of Professional Societies on the Advancement of the Profession: The Systems Engineering Example"*, Submitted and presented at the International Council on Systems Engineering (INCOSE) Conference, Chicago
3. June 2010, *"Cost and Risk Considerations for Testing Unmanned and Autonomous Systems of Systems"*,Submitted and presented at the IEEE Systems of Systems Conference, Loughborough, England
4. February 2009, *"Evaluation of Strategies to Reduce Taxi-Out Emissions at Airports*", Submitted to American Institute of Aeronautics and Astronautics
5. December 2008, *"Opportunities for Reducing Surface Emissions through Airport Surface Movement Optimization"*, Submitted to the Federal Aviation Administration

Co-Author

6. July 2010, *"Normative and Descriptive Models for Test & Evaluation of Unmanned and Autonomous Systems of Systems*", INCOSE Symposium, Chicago
7. February 2010, *"Unmanned and Autonomous Systems of Systems Test and Evaluation: Challenges and Opportunities"*, IEEE Systems Conference, INCOSE Track

Conferences Attended:

1. International Council on Systems Engineering Conference (Chicago - 2010) - Topic: The role of professional societies in the advancement of the profession: The Systems Engineering Example
2. IEEE Systems of Systems Engineering Conference (Loughborough, UK, 2010) - Topic: Cost and risk considerations for testing unmanned and autonomous systems of systems
3. COCOMO Forum (Los Angeles, 2010)- Topic: Cost elements and policy implications for testing unmanned and autonomous systems of systems
4. International Test and Evaluation Association Conference (Texas, 2010) Cost drivers for testing unmanned and autonomous systems of systems

Poster Presentations:

1. MIT, MSRP, (2008): Annual poster presentation on the topic: Control Algorithms for Synchronized Position Hold Engage and Reorient Experimental Satellites (SPHERES)
2. Revolution in Aviation, Palm Springs, CA. (2009) Topic: Emissions Reduction Strategies for Taxiing Aircraft
3. Lean Advancement Initiative, Annual Research Conference (2010): Topic: Cost and Risk Considerations for Testing Unmanned and Autonomous Systems of Systems

Acknowledgments

I am forever in debt to those persons who have contributed to the outcomes of this work. First, my family for instilling in me the value of education and constantly reminding me that I could accomplish whatever I set my mind to. Second, to my academic advisers, who have supported me through thick and thin, and third, to the organizations that have supported this research through funding, expertise and data.

To my dad, who could not be here to see the completion of this work, I want to say: Thank you, and I did it! I got that degree you wanted me to. To my mom, who was very patient and has done an excellent job teaching me the values to be a member of society. To my sisters, Chandra and Savitri, your support has meant the world to me. To my friends and confidants, you've listened to me every day, you've given me the comfort of knowing you were there as a source of support, and you pushed me to be all that I could be. I will be forever grateful.

The work in this thesis was fueled by my adviser's passion and dedication to research, and to see his students succeed. Ricardo Valerdi, you've done it. In my world, you are a leader by example, and that person who will always be my mentor, teacher, and friend. Debbie Nightingale, many times you were my knight in shining armor, coming to my rescue, when I thought all hope was lost. Without your support, this thesis would not be a reality. JK, I consider you my honorary adviser and friend and will be forever grateful for your support.

The realization of this effort would not have been possible without the assistance of the PATFrame team: Neno, George, Susan, Jenny, Ricardo, Kris, John, Bob, Tsoline and Gaurav. The data presented in this work was made possible by Mr. Joseph Lucidi, and I thank him for taking the time from his busy schedule to make this work a reality. I am also grateful to the Test

Resource Management Center for funding this work and for Tom Tenorio for going above and beyond to support this project.

This material is based upon work supported by the Department of Defense, United States Army, White Sands Missile Range, NM under Contract No. W9124Q-09-P-0230. The authors would like to thank the Test Resource Management Center (TRMC) Test and Evaluation/Science and Technology (T&E/S&T) Program for their support. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Department of Defense, United States Army, White Sands Missile Range, NM.

Chapter 1 – Introduction

"They're going to sneak up on us...They're going to do more and more of the toting. They're going to do more and more of the surveilling. And when they start fighting, no organized force could stand against them"

- John Pike, GlobalSecurity.org (Singer, 2009)

Government and private-sector interest in unmanned and autonomous systems (UASs) is growing, due in large part to the U.S. military's expanded development and use of these systems in Iraq and Afghanistan. The absence of a pilot or any other humans on board allows them to perform a variety of missions not generally considered favorable for manned systems. UASs can also perform dangerous missions without risking loss of life. UASs have been used for a number of years for various purposes, such as collecting scientific data, assisting with border security, providing and connecting communication networks, gathering weather data from inside hurricanes, fighting wars, and basically performing tasks and accessing environments which could pose a threat to humans. For example, in the aftermath of Hurricane Katrina, UASs searched for survivors in an otherwise inaccessible area of Mississippi and in 2004, the U.S. Geological Survey and the U.S. Forest Service used a UAS to study renewed volcanic activity at Mount St. Helens, Washington (United States Government Accountability Office, 2008).

Perhaps one of the most controversial topics in the deployment of UASs is the exponential growth in demand from the Department of Defense (DoD), and the constant challenge of ensuring that the systems that are delivered are safe and fit for operation. Some of the higher level risks of UASs include unintended or abnormal system mobility operation, inadvertent firing or release of weapons, engagement or firing upon unintended targets, self-damage of own system from weapon fire or release, personnel injury, equipment damage, environmental damage, system loss and system collision (Department Of Defense, 2007). However, although enumerating all possible routes to failure may sound like a simple task, it is difficult to exhaust all the alternatives. Usually a system must be modeled in different ways before analysts are confident that they have grasped its intricacies, and even then it is often impossible to be sure that all avenues have been identified (Morgan, 1993).

To make matters more complicated, systems today interact with one other and form a net centric entity in an integrated and well connected network which is referred to as systems of systems (SoS). This means that the constituent UASs are both operationally independent (most or all of the constituent systems can perform useful functions both within the SoS and outside of the SoS) and managerially independent (most or all of the constituent systems are managed and maintained by different decision makers) (DoD, 2008). So from here on, they will be referred to as Unmanned and Autonomous Systems of Systems (UASoS). In order to be useful, a UASoS must have the capacity for adaptation to change no matter what mission it has to perform. However, because these systems are so tightly coupled, the interconnected parts must be rigorously managed since their emergent behavior can be extremely complex. Addressing such issues requires a fundamental understanding of the risks associated with UASoS.

Motivation

UASoS provide new challenges, dictating very different developmental testing, which focuses on identifying technical capabilities and limitations, and operational testing, which is the decision maker for deployment. Currently, systems designed under traditional means are expected to perform predictable tasks in bounded environments and are measured against their ability to meet requirements, while UASoS function and operate in open, non-deterministic environments and are more focused on interactions between components, both manned and unmanned. The structure and demands for UAS performance have outgrown the capabilities of current test and evaluation (T&E) processes (Macias, 2008). Test has huge overhead and is highly optimized for yesterday's problems. Systems are becoming too complex – and this is further increased as human redundancy is being taken out of the loop- and there is more reliance on the performance of remotely operated machines. Varying and changing expectations create an environment of confusion throughout the acquisition process, and T&E is yet to adapt to these changes.

Forced to balance the need for practical programs against problems that do not seem to lend themselves to simple solutions, policy-makers could easily become mired in intractable, almost existential, dilemmas. There is need to focus now on how to anticipate the challenges that more complex systems pose, and how to develop a testing infrastructure that adapts to these types of challenges as they arise. Infrastructure does not only refer to test procedures, but also the processes, people and overall strategy of T&E. And because finding and fixing problems after delivery is often 100 times more expensive than finding and fixing it during the requirements and design phases, it is even more critical to focus on deciphering new ways of testing and focus more on mission, capabilities, and effectiveness (B. Boehm & Basili, 2001).

A UASoS requires the ability for manned and unmanned systems to co-operate with each other to fulfill its purpose. Many factors can increase the integration complexity of the SoS including the number of systems to be integrated, number of interfaces involved and technology maturity of the SoS. In addition, the number of requirements of the SoS is a key driver of risk, as well as changes in requirements throughout SoS development and operation. Many times it is unclear what the SoS needs to do in order to fulfill its mission and without the appropriate metrics to evaluate the performance of the UASoS, it is difficult to determine whether the mission is successful or not. Furthermore, not only do requirements change within a mission setting; missions and operational platforms also change resulting in changing requirements to reflect the warfighter's needs. A typical SoS integrates a number of operational platforms, and a versatile mix of mobile and networked systems that will leverage mobility, protection, information and precision. To conduct effective operations across such a spectrum requires careful planning and co-ordination of space, air, land domains that are connected by networks. Decision makers must also understand the SoS architecture and capabilities, as well as interoperability across all components of the SoS. Further, the individual systems within a SoS may have varying levels of maturity and may enter the SoS at different stages of the SoS lifecycle (Krygiel, 1999). Ensuring that these systems can still work together and merging newer more advanced technologies with more traditional technologies can present a significant challenge to development and validation of the SoS.

Morgan (1993) states that if there are inadequate approaches to assessing risks, this may result in bad policy. Unfortunately, such is the case existing for the deployment of UASoS. Testing at the SoS level requires focus on the interactions between the SoS constituents and the emergent behaviors that result from the complex interactions between the constituent systems (Dahmann, Rebovich, J. A. Lane, & Lowry, 2010). Current test procedures are not set up to determine what these interactions are and while infinite testing could potentially minimize every possible risk in every mission scenario, no program can afford such luxuries. Significant tradeoffs must be made in terms of cost, effort, and risks under uncertainty, especially with regards to the possible interactions between the systems. Currently, there is no standard method to determine what is really required to get programs to the point of safe deployment, nor is there the ability to begin making effective contingency plans should testing requirements change (Macias, 2008). It is possible that these problems face so much uncertainty, that pressures inevitably prompt action before enough information is gathered to establish a causal chain. Spending too much effort designing goals when causal understandings are still in flux is inefficient. As such, policy making and enforcing policies on the deployment of UASoS becomes very problematic.

Verification and validation, commonly referred to as testing, is required especially for UASoS to identify risk, improve capabilities and minimize unpleasant surprises. In many ways, while the risks of UASoS are still uncontrollable, testing makes these risks more observable, teasing out the issues that may arise by allowing the UASoS to react under various scenarios. To identify all the possible risks of testing would probably require an infinite supply to resources, time and labor. Unmanageable combinatorial problems can result when a large number of tests need to be performed on a large number of systems, and especially in the DoD, there is a need to prioritize tests to ensure the systems meet schedule requirements. The type of test and amount of each type of test to be performed will also be a driver of costs. For example, live tests require considerable resources, labor, and scheduling, and are significantly more costly than a simulated test which can be done in a virtual environment. While it is impossible to eliminate all risks through computer simulations; the more scenarios that can be recreated and tested in a simulated environment, the more failures that can be teased out before making a decision on whether more live testing is needed. Multisite coordination for testing also becomes an issue especially when multiple stakeholders are involved and individual systems are located in many different places. Testing systems in specific domains can also be difficult especially in the space and undersea arenas which are primarily UAS environments and access becomes logistically more difficult and expensive. Autonomy is also an important factor for test and evaluation of UASoS. Autonomous systems add an additional level of complexity because the performance of unmanned systems in scenarios that are not anticipated is difficult to replicate not only at the system level, but also at the SoS level. As individual UASs are merged with other systems to form a SoS, there is need for a better understanding of the risks associated with testing in multiple domains as well as the platforms necessary to ensure effective testing in space, air, land, sea and undersea domains at once. When systems are integrated, it is difficult to predict how the test process needs to adapt to account for emergent properties, especially when dealing with UASoS, as this places additional demands on limited resources and time. For example, if a program is critical to delivering a capability, testing needs to be efficient and effective enough to allow multiple increments so that programs have a chance of being fielded on time.

The Test Resource Management Center (TRMC) is the organization within the DoD responsible for setting policies for verification and validation activities ("WEAPON SYSTEMS ACQUISITION REFORM ACT OF 2009," 2009). Its charter is to plan for and assess adequacy and to provide adequate testing in support of development, acquisition, fielding, and sustainment of defense systems; and, maintain awareness of other T&E facilities and resources, within and outside the Department, and their impacts on DoD requirements (Tenorio, 2010). Through its established directives on testing, TRMC is providing a basis for determining whether a UASoS gets fielded or not, and whether it is allowed to keep progressing through the acquisition cycle. Current test planning procedures require that commercial testing and experience be recognized, all potential testing impacts on the environment be considered, full use of accredited models and simulations be adopted, and all technical capabilities and limitations of possible alternative concepts and design options be considered. However, more attention needs to be paid to the testing of UASoS because there is need for T&E processes to recognize levels of effectiveness, to focus on the interactions between components and emergent behaviors, and develop the ability to make effective contingency plans as requirements change.

Future for T&E

Within the TRMC, The Unmanned and Autonomous Systems Test group focuses specifically on UASs and recently UASoS. In a recent briefing it was established that "In any wartime situation, it is clear that the first priority is to develop and deliver solutions to the warfighter in order to reduce causalities and improve mission success. In many cases, urgent needs demanded that new capabilities or technologies be envisioned, developed, manufactured and shipped to units in the field without any testing or training – and in many cases this was justified as a quick reaction. Such approach, however, is only effective if testing and training are done in parallel in an expedited fashion" (Tenorio, 2010)

Testing needs to be effective and focused, determining the issues, working towards ensuring the risks are known and determining ways of minimizing them. To identify and address the technical risks, it is required that the UASoS be stressed beyond their perceived normal operational limits to ensure the robustness of the design in varying operational environments, and that all weapon, information, command, control, communications, computers, intelligence, surveillance, and reconnaissance programs that depend on external information sources, or that provide information to other DoD systems, are tested and evaluated for information assurance (DoD, 2008). It is also necessary to have adequate feedback loops, a culture of information sharing and learning from best practices, as well as the development of metrics and/or performance indicators that adequately reflect the ability of the test process to meet the expectations of the programs.

The reality is that policies must be chosen from a proliferation of incomplete information that relates possible policy actions to outcomes. These policies will likely endure for years, even decades, during which time the available information will likely improve. Faced with ambiguous evidence, incomplete expert understanding of the underlying causal chain in question and even a lack of reliable indicators, decisions must nevertheless be made and justified. What is now needed is a testing infrastructure that helps fill the gaps of lack of information, best practices, and ability to adapt to changes as UASoS become more complex. Testers and evaluators have much work to do develop test procedures, develop test facilities, and develop evaluation methods and criteria to address the unique characteristics, operation, and missions of UASoS. Risk managers can help by setting up an infrastructure that identifies the risks more effectively and working to prevent the processes producing the risk, to reduce exposures to modify effects, to alter perceptions or valuations through education and training. Decision frameworks must be carefully and explicitly chosen and that these choices are kept logically consistent, especially in complex situations. To do otherwise may produce inconsistent approaches to the same risk (Morgan, 1993).

The Prescriptive and Adaptive Testing Framework (PATFrame), currently under development, uses knowledge acquisition to minimize risk through a decision support system (Hess, Cowart, Deonandan, Kenley, & Valerdi, 2010). Under this framework, the word *prescriptive* refers to a decision assessment that involves suggestions of appropriate decision behavior that can lead to the best outcomes. Under a purely normative framework, decisions are made through rationale. In formal approaches, a set of *axioms* that a rational person would surely agree is postulated, which leads to the *normative* or most desirable or optimum behavior a decision maker should seek to follow. The normative approach defines how test and evaluation should be completed and is stipulated through standards and instructions that state what needs to be accomplished in order for a system or capability to be adequately tested. However, how human beings react in real situations and actually make decisions reflects the descriptive method of decision making, and is determined by actual experiences. In DoD test and evaluation, this would apply to how an actual test mission is planned or takes place, which might not be the specific normative way of planning test missions. Prescriptive is meant to provide direction in order to apply a correction caused by a deviation from the norm based on the actual behaviors of people. A methodology for the test and evaluation of UASoS needs to be developed in order for stakeholders in the DoD Test and Evaluation enterprise to obtain their maximum value from these types of missions.

In addition, on March 25, 2010, Donald Macwillie, Brigadier General of the United States Army Operational Test Command, released a memorandum entitled "Test Cost Estimates" (Macwillie, 2010). He specified that:

1. Test costs are a crucial element of operational testing. As an organization, we must continue to be stewards of public resources and provide other agencies that work with us the ability to plan and execute testing with as much transparency as possible.

2. I acknowledge that test costs change as test requirements are refined and finalized. Test directors are in the best position to use their experience and military judgment to assess the impact of changes and associated costs. I expect

test directors to review test costs with the same rigor that I do to ensure good stewardship, improved estimating, and transparency to customers.

3. To accomplish this coordinated effort, test directors will approve all test cost estimates.

The increasing frequency and number of programs that have run significantly over-budget and behind schedule because T&E problems were not adequately understood should, by itself, be reason enough for the acquisition community to press for improvement in forecasting T&E resource needs. This, coupled with DoD budget restructuring and cuts, has forced many to reconsider how they operate on a daily basis, plan in advance, and deal with the consequences of their actions. On September 14, 2010, Dr. Ashton Carter, the current Secretary of Defense, released a memorandum focused on "Better Buying Power: Guidance for Obtaining Greater Efficiency and Productivity in Defense Spending", in which he emphasized the "do more without more" principle. Program managers now need to treat affordability as a key performance parameter in an effort to conduct a program at a cost constrained by the maximum resources that the department can assign to the capability, which requires programs to use methods to minimize their cost and schedules as effectively as possible. Further, a "should cost" analysis at the beginning of the program requires early value proposition for each element of the program with an evaluation at the completion of each milestone set forth at the beginning. A "Fixed Cost" approach also helps align objectives and make projects less expensive when the government is clear on what it wants from the beginning, does not change its mind and when industry has good control of its processes and costs to name a price (Carter, 2010).

Thesis Statement

This work seeks to understand the cost and risk considerations for UASoS T&E and propose the development of a parametric cost model to conduct trade-off analyses for T&E within the PATFrame decision support system. A risk and cost approach is used because it is recognized that on a SoS level, there must be a comprehensive analysis of complexity to understand its impact on the cost of systems and to avoid unreliable estimates and unfavorable system performance. This process can also produce strategic options to improve the confidence of cost estimators and stakeholders in making better decisions, even in the face of complexity, risk, and uncertainty (Dixit & Valerdi, 2007). Developing any cost or resource estimation model for T&E requires a fundamental understanding of existing cost estimation techniques, how they have evolved over the years and how they can be leveraged for the purpose of T&E of UASoS. This thesis focuses on understanding the need for better estimation of the test effort for UASOS, what cost and risk considerations must be addressed specifically for the UASoS T&E and how other approaches may be limited in addressing the specific issues of T&E of UASoS. The work presented here is a combination of information collected from various normative and descriptive views of testing based on literature review, surveys, and interviews with members of the DoD community. Information presented represents the initial stages of identifying specific parameters for the development of the cost model and provide management guidance to the DoD T&E community in estimating the effort required for test and evaluation of inter-related unmanned and autonomous systems in the context of SoS.

Thesis Roadmap

Chapter 1 of this thesis introduces the concept of UASoS, highlights some of the challenges as UASoS progress and the motivation for a new testing infrastructure that includes better cost estimation techniques. Chapter 2 focuses on various existing cost estimation techniques and previous work done in cost estimation both within the DoD and beyond. It also highlights areas that can be leveraged for the cost estimation approach presented in this work. Chapter 3 talks about the methodology and tools used to conduct research and build the model. Chapter 4 defines the model and describes the main parameters and variables included in the cost model. Chapter 5 illustrates the results of a data collection case study, and Chapter 6 summarizes the future implementation of the cost model. The final chapter focuses on the implications of such a model for UASoS T&E, and what is needed in order to further develop and adopt the cost model into the current test infrastructure.

Chapter 2 – Background and Related Work

An important part of developing a model such as one for UASoS T&E is recognizing previous work in related areas. This process often provides a stronger case for the existence of such a model and ensures that its capabilities and limitations are clearly defined. In this section, an overview of the existing cost estimation techniques is given, their advantages and disadvantages are identified, and a case is made for developing a cost model for UASoS T&E.

Overview of Cost Estimation Techniques

A number of cost estimation approaches currently exist, varying both in maturity and sophistication. Some are more easily adaptable to changing and emerging environments, whereas others take more time to develop. While the logic behind each of these approaches are fundamentally different, leaving only their results as measures of merit, it is believed that a hybrid approach that combines these techniques is the best way to capture the effort for UASoS T&E that a single approach may overlook. Each technique has its advantages, but it also has disadvantages in estimating cost especially as systems become more and more complex. Some of these techniques are presented here.

Analogy/Comparative/Case Based Reasoning:

This technique requires comparing available data from similar completed projects, and adjusting estimates for the proposed project. This allows organizations to capitalize on memory and experience, as opposed to reinventing the wheel every time a new project comes along. Case studies represent an inductive process, whereby estimators and planners try to learn useful general lessons by extrapolation from specific examples. They examine in detail elaborate studies describing the environmental conditions and constraints that were present during the development of previous projects, the technical and managerial decisions that were made, and the final successes or failures that resulted. They then determine the underlying links between cause and effect that can be applied in other contexts. Ideally, they look for cases describing projects similar to the project for which they will be attempting to develop estimates and apply the rule of analogy that assumes previous performance is an indicator of future performance. The sources of case studies may be either internal or external to the estimator's own organization (Valerdi, 2005). They have the advantage of being reliant on historical data, being less complex than other methods, and saving time. However, there may be subjectivity and bias involved, may be limited to just mature technologies, and sometimes rely on a single data point. It can also be difficult to identify the appropriate analogy, and there is also the risk of applying linear analogies to non-linear systems, especially as systems become more complex (Young, Farr, & Valerdi, 2010).

Expert opinion

This is produced by human experts' knowledge and experience via iterative processes and feedbacks and is the most informal of the cost estimation techniques because it simply involves querying the experts in a specific domain and taking their subjective opinion as an input. A Delphi method is used to capture the opinions of the experts and is explored more in the Methodology Section. Especially where there is insufficient empirical data, parametric cost relationships, or unstable system architectures, this approach is useful and a very simple fallback. However, it is seductively easy. The obvious drawback is that an estimate is only as good as the expert's opinion, which can vary greatly from person to person, not to mention the fact that years of past experience does not guarantee future expertise as requirements change, systems change and become more complex. Further, even the experts can be wrong and highly subjective and biased. Detailed cost drivers may be overlooked and program complexities not fully understood can make estimates less reliable.

Top Down & Design To Cost:

This technique is based on the overall project characteristics and derived by decomposing into lower level components and life cycle phases. It is very system oriented, with minimal project detail required and leads to fast and easy deployment. Once a total cost is estimated, each subcomponent is assigned a percentage of that cost. The main advantage of this approach is the ability to capture system level effort such as component integration and configuration management. However, the top down approach can often miss the low level component details and major cost drivers that can emerge in large systems (Young, Farr, & Valerdi, 2010). It also lacks detailed breakdown of the subcomponents that make up the system and can therefore lead to limited detail available for justification.

Bottom Up & Activity Based Approach:

This is opposite to the top-down approach, and begins with the lowest level cost component and rolls it up to the highest level for its estimate. The estimate is made directly at the decomposed component level leading to a total combined estimate. This method is sometimes referred to as "Engineering Buildup" and is usually represented in the form of a Work Breakdown Structure (WBS), which makes the estimate easily justifiable because of its close relationship to the activities required by the project elements. At a lower level, this can be a fairly accurate estimate since the estimate is usually provided by the people who will be doing the actual work. However, this method relies on stable architectures and technical knowledge (Young, Farr, &

Valerdi, 2010). The process involved is very labor, data and time intensive and can thus be very expensive and inconsistent depending on the application. It may even result in overlooking integration costs, and lacks the ability to capture economies of scale. Further, because of the various layers, it is easier to double count expenses from one level to the next, which can result in overestimates.

Actual Costs/ Extrapolation Method:

This method uses costs experienced during prototyping, hardware engineering development models and early production items to project future costs for the same system. It is able to provide detailed estimates, and rely on actual development data. However, the development may not always reflect cost correctly and there is a high degree of uncertainty related to what the actual cost should be based on how the extrapolations are made. It is also heavily dependent on actual existing data which may be unavailable at the time the estimate is needed, and may also require various levels of detailed involvement (Young, Farr, & Valerdi, 2010).

Parametric Cost Estimation Models:

A parametric cost estimation model is defined as a group of cost estimating relationships (CERs) used together to estimate entire cost proposals or significant portions thereof. These models are often computerized and may include many interrelated cost estimation relationships, both cost-to-cost and cost-to-non-cost. Parametric models generate cost estimates based on mathematical relationships between independent variables (i.e., requirements) and dependent variables (i.e., effort). They use mathematical expressions and historical data to create cost relationships models via regression analysis. The inputs characterize the nature of the work to be done, plus the environmental conditions under which the work will be performed and delivered. The definition of the mathematical relationships between the independent and dependent variables is the heart of parametric modeling. (Valerdi, 2005). These CERs are statistical predictors that provide information on expected value and confidence, have less reliance on systems architectures and are less subjective since they incorporate data from a number of similar past projects. However, this can also be a disadvantage because there is a high reliance on historical data, and the attributes within the data may be too difficult to understand. Further, they can be very resource intensive, especially investing time and labor in developing cost drivers, collecting data, and then developing the CERs based on these data. Reliable data is crucial to this type of cost estimation and data can be very difficult to collect based on people availability and past data documentation available. As such the development of any CER is limited to the data availability and variables identified through the process (Young, Farr, & Valerdi, 2010).

UASoS T&E Cost Model Lineage

The undeniable trend is toward increasingly complex systems of systems dependent on the coordination of interdisciplinary developments where effective testing is no longer just another phase in the acquisition life cycle, but the key to ensuring the safety of all stakeholders especially users and innocent bystanders. It is known that increasing front-end analysis reduces the probability of problems later on, but excessive front-end analysis may not pay the anticipated dividends or address the key issues which should be a priority. The key is to accurately estimate early in a program the appropriate level of test effort required in order to ensure system success within cost and schedule budgets, as well as ensure that UASoSs are adequately tested to ensure safety.

The use of parametric models in planning and management serves as valuable tools for engineers and project managers to estimate effort. While cost models have not been specifically applied to testing and evaluation in the past in the DoD, they have been an essential part of DoD acquisition since the 1970s. Hardware models were first to be developed and were followed by software models in the 1980s. The early 1980's marked an important stage in the development of a parametric community of interest, including conferences such as the Association for Computing Machinery Special Interest Group on Metrics and the Forum on COCOMO and Systems & Software Cost Modeling; journals such as Cost Engineering Journal, IEEE Transactions on Software Engineering, and Journal of Cost Analysis and Management; and books such as Boehms' "Software Economics" and "COCOMO II". These included the refinement of earlier models such as PRICE S and SLIM, and the development of early-1980's models such as SPQR/Checkpoint, ESTIMACS, Jensen/SEER, Softcost-R, and COCOMO and its commercial implementations such as PCOC, GECOMO, COSTAR, and Before You Leap. These models were highly effective for the largely waterfall-model, build-from-scratch software projects of the 1980's and defined the early achievements of the field of parametrics (Valerdi, 2008).

The 1985-1995 time period primarily involved proprietors of the leading cost models addressing problem situations brought up by users in the context of their existing mainstream capabilities. Good examples are the risk analyzers, either based on Monte Carlo generation of estimate probability curves, or based on agent-based analysis of risky combinations of cost driver ratings. Between 1995 and 2005, the improvement of existing parametric models was based primarily on the realization that the underlying assumptions of the existing models were based on sequential waterfall-model development and software reuse with linear savings were becoming obsolete. The projection of future hardware components also shaped the development of several new parametric models.

Various cost models have subsequently been developed to focus on specific categories of systems; however none of them have been singled out for the testing and evaluation phase of the system life cycle. In fact, previous studies on systems engineering cost models have shown that developers are so convinced that T&E is such a small proportion of the total life cycle cost, that much more emphasis is placed on the cost of the other phases of the life cycle as opposed to T&E (Valerdi & Wheaton, 2005). However, further analysis of T&E in the SoS environment with recent reports of unexplained behaviors in complex systems (e.g., Lexus cars speeding out of control) are leading experts to re-evaluate these ideas (J. Lane & B. Boehm, 2006).

From a warfighters' perspective, testing UASoS is absolutely critical and in fact because many of these systems are being fielded for the first time and testing is so integrated with both development and operations, T&E contributes significantly to the cost of the system especially given the risks and uncertainties associated with UASoS. The budget, both in terms of cost and effort, is currently determined based on similar projects that have been conducted in the past, coupled with extrapolations to account for the new system under test. However, UASoS do not have a significant history, but are in such high demand that there is the need to understand how much effort is required for testing. Testing is often reduced to a purely technical issue leaving the close relationship between testing and business decisions unlinked and the potential value contribution of testing unexploited (Q. Li et al., 2009). There comes a point at which the amount of effort invested does not minimize risk at a justifiable rate. Neither does it offer enough of a return on the amount of resources invested into the test.

Today, there are fairly mature tools to support the estimation of the effort and schedule associated with UASoS T&E. For software development activities, there are the COCOMO II, Cost Xpert, Costar, PRICE S, SLIM, and SEER-SEM cost models. At the single system level, there is the Constructive Systems Engineering Model, COSYSMO, to estimate the system engineering effort and for definition of the SoS architecture, the solicitation and procurement process for the SoS components, and the integration of the SoS components into the SoS framework there is the Constructive System-of-Systems Integration Cost Model, COSOSIMO (J. Lane & B. Boehm, 2006).

But, while COSOSIMO addresses the development of a SoS and normative integration and testing in the SoS environment, there has been little work done with respect to the needed evolution of SoS T&E (prescriptive) or the evaluation of the flexibility and emergent behaviors of complex systems and SoS (adaptive limits). How do you know when testing is done and you have minimized sufficient risk so that the SoS is safe for deployment in the field? Li et al propose a value-based software testing method to better align investments with project objectives and business value (Q. Li et al., 2009). This method could provide decision support for test managers to deal with resource allocation, tradeoff and risk analysis, and time to market initiatives and software quality improvement and investment analysis. While Li's value based testing techniques do give a good foundation on which to build a methodology for a cost model for UASoS T&E, this method is more applicable for business critical projects focused on return on investment and not suitable for safety critical domains. It also requires detailed cost estimation to assist the test planner and does not account for emergent properties as those frequently found in UASoS. From a warfighter's perspective, a risk based testing approach may be more relevant as it focuses resources on those areas representing the highest risk exposure. Li also applies a costing methodology which defines costs of tests relative to each other as opposed to the absolute cost of test. PATFrame methodology attempts to calculate the absolute cost of test rather than relative cost because this will allow us to estimate and predict what strategies can be used to optimize the test process on a case by case basis.

In a paper entitled "Managing your way through the integration and test black hole", George also tries to address both testing and integration from a software perspective (George, 2010). She claims that the integration testing phase is a black hole, which the systems never seem to escape. George calculates integration effort as a product of the number of predicted defects and the average time to find and fix a defect plus the product of number of test cases and the average time to run a test case. While this is a very simple model and could be expanded to other phases of a life cycle as opposed to just software testing, it assumes that the main problem with integration testing is defects. However, this methodology is insufficient in considering UASoS T&E as using only defect analysis can be very limiting since there are a number of other cost drivers which define the stopping point of a test. In fact, in a recent workshop, representatives from the army indicated that "defects" are not of that much of a concern in the SoS environment, but rather identification and evaluation of emergent behaviour is of more importance.

George also assumes that these defects are known, can be easily found, and that the investigator can estimate the amount of effort to remove the defects. For UASoS T&E, it is necessary to not only be able to identify and understand these single-system defects but also to have a firm grasp of the risks involved in integrating multiple UAS to form a complex system of systems, and determine the cost drivers associated with those risks.

In addition, the fundamental methods presented by Aranha and Borba to include the complexity and sizing of tests for UASoS, can be expanded upon. Their work attempted to estimate the size of a software test which is required to determine the test execution effort. This is because test managers have difficulties using existing cost models, since the effort to execute tests are more related to the characteristics of the tests rather than characteristics of the software. Their method focuses on using the specifications of the test to determine the size and complexity, which is used as an input for test execution effort estimation models (Aranha & Borba, 2007). Such methodology is very relevant to this work because as a UASoS increases in size so does the testing complexity and thus the required test effort. This research focuses on the UASoS and presents a methodology to calculate the test effort based on the complexity of the SoS.

However, Aranha and Borba define test size as the number of steps required to complete the test, complexity as the relationships between the tester and the tested product. From A UASoS T&E perspective, many more factors need to be taken into consideration to determine the size and complexity of the effort. These range from the number of requirements of the SoS, to the interactions between individual systems, individual systems at various levels of maturity, operation platform diversity, maturity level of the test given emergent UASoS, etc. There are also organizational factors that can increase the complexity of the interactions between systems, including understanding of the integration requirements depending on how well defined they are, the number of organizations or individual stakeholders managing the systems, understanding the overall architecture of the SoS, etc.

These challenges and potential size and cost drivers are explored in the following sections, and a methodology that builds on the works mentioned in this section is presented to determine the effort required for UASoS T&E.

Chapter 3 – Methodology

Parametric cost modeling requires an extensive data base of historic cost and performance data, assumes historical cost relationships will continue to hold true for future projects, and uses regression analysis as the fundamental tool for development. The parameters can be thought of as characteristics, and calculate cost as a function of physical and performance characteristics. The parameters are used to develop cost estimating relationships (CERs), using explanatory variables from a set of sample points which realistically reflect typical delays, problems, mistakes, redirection and changing characteristics of the phenomenon being measured. In aircraft development, examples of such variables include empty weight, speed, wing area, power, range, schedule etc.

In general, when developing these CERs, one first needs to determine potential "causes" of cost for each cost element, question the experts, and identify the potential cost drivers related to areas such as technology, size, performance and people. Then the functional forms of the relationships are specified. These must make sense, must be able to obtain good predictions rather than good statistics, and the shape of the line should not be determined by the data unless there is a lot of it. It is also important to ensure that cost behaves as expected when the cost driver varies. This process is heavily dependent on data, and analogous systems need to be carefully chosen to get quality data to use in building and calibrating the model.

The methodology adopted for this work is a combination of field research and quasi-experimental research. A combination of these approaches is used because each has its strengths that provide significant benefits because they use different perspectives in collecting data and also because having the right frame of mind while defining the hypotheses and then testing them is very important. The nature of the research question – how to estimate the effort for UASoS T&E – played the major role in determining the selection of these approaches.

Research Design

The purpose of field research design is to study the background, current status, and environmental interactions of a given social unit. In the context of this research, this refers to the DoD and its contractors, who are responsible for ensuring that all SoSs are sufficiently tested and safe for the warfighter. The expert data was collected through Delphi Surveys and interviews, to understand how testing is currently done, what some of the potential drivers of cost are, and how these impact the effort needed for UASoS T&E. Field research is useful because it provides:

- An in-depth analysis of current T&E organizations and personnel
- Useful examples to illustrate more generalized statistical findings
- Observations of real world activities and how these relate to theory

Quasi-experimental research design is used to approximate the conditions of the true experiment in a setting that does not allow control or manipulation of all relevant variables since factors that affect the conditions can compromise the validity of the design. It looks like an experimental design but lacks the key ingredient - random assignment. In UASoS T&E a number of organizations are involved, all of which are influenced by multiple outside forces including bureaucratic culture, politics, customer pressures, budget constraints, technical obstacles, mission priorities, critical issues etc, and it is impossible to control all of these conditions. The quasi-experimental research design is useful because it allows the:

- Investigation of cause-and-effect relationships
- Variance of different types of efforts in different conditions

- Opportunity to test various hypotheses

Research Approach

To derive good cost estimating relationships from historical data using regression analysis, one must have considerably more data points than variables; such as a ratio of 5 to 1 (Valerdi, 2005). It is difficult to obtain actual data on testing and evaluation costs and the factors that influence these costs especially when programs of record do not exist. Therefore, the Seven Step Modeling Methodology created by Barry Boehm and used for a number of cost estimation models (B. W. Boehm et al., 2000), was used in this research.

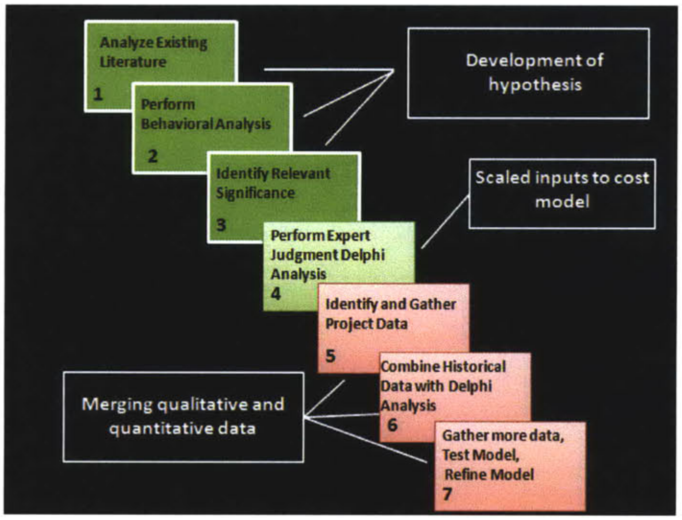


Figure 1: The Boehm Seven Step Modeling Methodology

For Steps 1 and 2, the interpretivist approach, which focuses on complexity of human sense making as the situation emerges, was used. This allowed the investigator to learn as much as possible about UASoS T&E and arrive at qualitative conclusions as to the most important factors. The interpretivist approach was used when developing the size and cost driver definitions with the PATFrame group and affiliates. Part of this effort also involved understanding how T&E is currently done across the services, determining the similarities and differences, understanding the language used when describing UASoS T&E and coming up with a generalized way of describing the T&E process across the services. Through a series of interviews, surveys, and working group meetings, the most significant drivers of cost were identified and defined, and a work breakdown structure that highlights the main activities involved in the T&E process, was created. The criteria for the interpretivist approach revolves around ensuring that there was credibility in establishing a match between the constructed realities of UASoS T&E and the respondents, and confirming that these cost drivers were grounded in the theory of cost estimation as well as testing and not just a product of the imagination.

Once the drivers were defined, there was a shift in the research strategy to a positivist approach. The positivist approach is used in steps 3, 4, 5, 6, and 7 because they involve the validation of the hypotheses. This approach focuses on making formal propositions, quantifiable measures of variables, hypothesis testing, and the drawing of inferences about a phenomenon from a representative sample to a stated population. This helps to construct validity in establishing the right measures for T&E size and cost, ensures internal validity establishing a causal relationship between the drivers and T&E effort, external validity establishing a domain in which these drivers can be generalized for T&E, and reliability in ensuring that these relationships between size and cost can be repeated in varying situations with the same results. The shift from the interpretivist to the positivist approach is analogous to a shift from the qualitative to the quantitative approach. Table 1 shows how the research design and approaches are related to each step in the methodology used.

Table 1: Research Designs and Approaches Used in the Boehm 7 Step Modeling Methodology

	Step 1: Analyze Existing Literature	Step 2: Perform Behavioral Analysis	Step 3: Identify Relative Significance	Step 4: Perform Expert Judgment Delphi Assessment	Step 5: Gather Project Data	Step 6: Determine Bayesian A-Posteriori update	Step7: Gather More Data, Refine Model
Field Research Design		X		X	X		X
Quasi-experimental Research Design			X			X	X
Interpretivist Approach	X	X					
Positivist Approach			X	X	X	X	X

Data Collection

Steps 4, 5 and 7 involved data collection. Expert data was collected in Step 4 and historical data was collected in Steps 6 and 7. A Delphi survey was used in Step 4. Developed at The RAND Corporation in the late 1940s, it serves as a way of making predictions about future events - thus its name, recalling the divinations of the Greek oracle of antiquity, located on the southern flank of Mt. Parnassus at Delphi. More recently, the technique has been used as a means of guiding a group of informed individuals to a consensus of opinion on some issue. Experts involved in UASoS T&E were first surveyed and asked for their opinions on the initial technical and organizational risks initially identified as factors that could potentially contribute to the cost of T&E. Those who were surveyed had been involved in the T&E process for at least 10 years, either as a tester, test engineer, test planner, evaluator or program manager. Respondents were asked to rate the identified risks on a scale of 1 to 5, with 5 having the greatest impact on the effort required for UASoS T&E and 1 having the smallest impact. This input was gathered help prioritize risks associated with UASoS T&E and define cost drivers, which are a combination of factors affecting SoS, individual systems and the testing process. The initial survey used to collect risk prioritization information is provided in Appendix A.

Data for historical projects were solicited from program managers who consulted with test engineers, testers and test planners to provide information. These program managers have at least 15 years experience in testing and are very knowledgeable of the test process. The data collection form, which was created in Excel and snapshots of which are shown in Appendix B, consisted of 5 sections. Section 1 provided instructions for the respondents including reference to the additional reference document. Section 2 asked for general information on the characteristics of the UASoS as well as the T&E process used and general outcomes of the effort. A generalized work breakdown structure was created and presented in Section 3, to further characterize the T&E process. Sections 4 and 5 asked respondents to provide quantitative data on the size drivers to help quantify the test effort, and rate cost drivers of the T&E effort. These surveys and data collection forms were designed for maximum measurement reliability by ensuring use of closed and open ended questions, knowledge questions to screen out respondents without enough information to answer the question, consistent measurement scales for questions of the same types, more than sufficient time to provide responses, question difficulty was consistent with the expertise of the respondents, and all forms were as short as possible to avoid repetition and cover only key and relevant points. Particularly for the data collection form, all extra information such as definitions were removed from the main tables and included in a separate reference document and hidden comments, so that the data collection form was relatively simple for respondents.

One of the challenges surrounding this research was the ability to collect data to define a fully calibrated model. One complete data set was collected and is presented as a case study in this work. The following sections describe the model developed in more detail, the potential for continued model development and implications for a new infrastructure in UASoS T&E.

Chapter 4 – Model Definition

From the beginning of this effort, this model has gone through several developments. The model assumes that the effort required for UASoS T&E is a function the program size, cost drivers, scale factors, and calibration constants. Each of these parameters has to be quantified using a combination of qualitative and quantitative methods described in the previous section. This effort follows the model form of COSYSMO (Valerdi, 2005) and the general form of the model is shown below.

$$PM = A * \underset{\overset{\nearrow}{\text{ADDITIVE}}}{(Size)}^{E} \underset{\overset{\uparrow}{\text{EXPONENTIAL}}}{} * \underset{\overset{\nwarrow}{\text{MULTIPLICATIVE}}}{(EM)}$$

(Equation 1)

Where:

PM = Person Months

A = calibration factor

Size = measure(s) of functional size of a system having an additive effect on UASoS T&E effort.

E = scale factor(s) having an exponential or nonlinear effect on UASoS T&E effort

EM = effort multipliers that influence UASoS T&E effort

The general rationale for whether a factor is additive, exponential, or multiplicative comes from the following criteria (Barry Boehm, Valerdi, J. A. Lane, & Brown, 2005).

1. A factor is additive if it has a local effect on the included entity. For example, adding another source instruction, requirement, test, interface, mission, operational scenario,

or system to the UASoS would create additive effects. The impact of adding a new item would be inversely proportional to its current size. For example, adding one test to the UASoS to one with 10 existing tests corresponds to a 10% increase in size while adding the test to a system with 20 tests would be a 0.05% increase.

2. A factor is multiplicative if it has a global effect across the overall UASoS T&E effort. For example, adding a test site, or an incompatible tester has mostly global multiplicative effects. Another example is in the case of autonomy. If a highly autonomous/intelligent system is added to a UASoS with 5 existing unmanned systems, this could increase the effort by 50%. Similarly, if this same autonomous system was added to a UASoS with only 2 existing unmanned systems, this could still increase the effort required by 50%.
3. A factor that is exponential has both a global effect and an emergent effect for larger UASoSs. If the effect of the factor is more influential as a function of size because of the amount of rework due to architecture, risk resolution, team compatibility, or readiness for UASoS integration, then it is treated as an exponential factor.

Model Form

$$PM_{NS} = A \cdot (Size)^{E} \cdot \prod_{i=1}^{n} EM_{i} \qquad \text{(Equation 2)}$$

Where:

PM_{NS} = effort in Person Months (Nominal Schedule)

A = calibration constant derived from historical project data

Size = determined by computing the weighted sum of the size drivers

E = represents economy/diseconomy of scale; default is 1.0

n = number of cost drivers

EM_i = effort multiplier for the i_{th} cost driver. Nominal is 1.0. Adjacent multipliers have constant ratios (geometric progression). Within their respective rating scale, the calibrated sensitivity range of a multiplier is the ratio of highest to lowest value.

Each parameter in the equation represents the Cost Estimating Relationships (CERs) that were defined by experts. The *Size* factor represents the additive part of the model while the *EM* factor represents the multiplicative part of the model. Specific definitions for these parameters are provided in the following sections. A detailed derivation of these terms can be found in Valerdi's derivation of the COSYSMO – Systems Engineering Cost Model (Valerdi, 2005). The dependent variable is the number of UASoS T&E person months of effort required under the assumption of a nominal schedule, or PM_{NS}. The derivations for each of these parameters require a significant amount of historical project data, which unfortunately, was not possible with this research. This study collected only one complete set of data, which is presented as a case study, while the specific size and cost drivers developed are explained in the following sections.

Size Drivers

Size drivers are used to capture the functional size of the UASoS under test. They represent a quantifiable characteristic that can be arrived at by objective measures, i.e. physical size of the SoS test effort. Intuition dictates that carrying out the test and evaluation for a combination of space, air, land, sea and undersea systems represents a larger effort than the test and evaluation of a subset of these domains. In order to differentiate between these types of UASoS, seven properties were developed to help quantify the difference, as well as reflect the current T&E practices used in the DoD. These include *# of SoS Requirements/Expectations, # of*

Mission Scenarios, # of Critical Operational Scenarios, # of Measures of Effectiveness, Performance and Suitability, # of Systems in the SoS, # of SoS Interfaces, # of Tests and *# of stakeholders involved.* These size drivers are quantitative parameters that can be derived from project documentation. Each size driver has both continuous and categorical variable attributes. As a continuous variable it can represent a theoretical continuum such as "requirements" or "interfaces", which can range from small systems to very large systems of systems; with most cases falling within an expected range. As a categorical variable it can be represented in terms of discrete categories such as "easy", "nominal" or "difficult" that cannot be measured more precisely. The assumption here is that "easy" size drivers would have less of an impact on cost as the "difficult" ones, which will be reflect in the total cost calculation. The definitions of the drivers and categorical attributes were determined through interviews and surveys and are presented in this section.

Three main factors influence size drivers, and are used as adjustment factors in cost estimation models. They are volatility, complexity and reuse. The test environment is a dynamic environment, which can create changing requirements, systems, test needs, interfaces, scenarios may change as requirements change, and the level of volatility can vary. New requirements can be created, new systems may be introduced, additional tests planned etc. Any volatility which is beyond what is expected and adjusted for in the size driver, can greatly contribute to an increase in size. Complexity can also vary among drivers, for example, requirement complexity can vary depending on how well they are specified, how easily they are traceable to their source, and how much they overlap there is. Typically a more complex requirement would have a higher weight assigned to it. The third factor, reuse, facilitates the usage of certain components in the T&E process and tends to bring down the efforts involved in the system development. However reused components may also require some effort of rework which will contribute to the overall cost of the project. For example, during test efforts, systems are reused for testing purposes, older systems merged with newer ones and while these have been used there is some work required to make them compatible with each other. Also, tests are reused from one test scenario to the next, and there is some expertise already gathered to minimize effort, but at the same time there is some rework to make the test adaptable to the new UASoS. In summary, volatility and complexity increase the size, whereas reuse has the effect of either increasing or decreasing the size of the UASoS T&E effort. For an explanation of more detailed impact and how these are dealt with in current cost estimation models, see Valerdi's dissertation, "A Constructive Systems Engineering Cost Model" (Valerdi, 2005).

1. Number of SoS Requirements/Expectations

It is very important to understand what the expectations of the UASoS are in order to design a test process that makes sure it meets those requirements. The number of SoS requirement/expectation can be found by counting of the number of applicable shalls/wills/should/mays in the SoS specification documentation. It is important to have a well defined boundary of the UASoS of interest, understand what the expectations are at each level, and determine the best way to decompose overall T&E objectives into these requirements without double counting. Lower level requirements should be disregarded if they do not influence the T&E effort.

Table 2: Number of SoS Requirements/Expectations Definition

<u>*Number of SoS Requirements/Expectations*</u> *This driver represents the number of expectations for the SoS-of-interest during the test phase. The quantity of expectations includes those related to the effort involved in testing the SoS and is a combination of the interface requirements, individual system requirements, and mission scenario requirements. These requirements may be functional, performance, feature, or service-oriented in nature depending on the methodology used for specification.*

Table 3: Number of SoS Requirements/Expectations Rating Scale

Easy	Nominal	Hard
Simple to implement	Familiar	Complex to implement or engineer
Traceable to source	Can be traced to source with some effort	Hard to trace to source
Little requirements overlap	Some overlap	High degree of overlap
Timelines not an issue	Timelines a constraint	Tight timelines through scenario network
Easy to map to test objective	Can be mapped to test objective	Cannot map to test objective easily

2. *Number of Mission Scenarios*

The mission scenarios are derived depending on the UASoS expectations. When a UASoS is assigned for testing, the testing personnel must coordinate with the test planner, users and program manager to determine what the appropriate mission scenarios will be and document this for further development. These mission scenarios are then broken down in the critical operational scenarios associated with each mission scenario. A count of mission scenarios can be made from the number of possible mission types that the UASoS has to perform, groups of tests geared towards various mission types, distinct use cases each with clearly defined inputs, outputs and processes found in the test plans and test reports.

Table 4: Number of Mission Scenarios Definition

Number of Mission Scenarios *This driver represents the number mission scenarios derived from the different capability requirements/expectations of the SoS. It shows the main operational concepts and interesting or unique aspects of operations. It describes the interactions between the subject architecture and its environment, and between the architecture and external systems.*

Table 5: Number of Mission Scenarios Rating Scale

Easy	Nominal	Difficult
Well defined	Loosely defined	Badly defined
Loosely coupled	Moderately coupled	Tightly coupled or many conflicting requirements
Few, simple off-nominal threads	Moderate number or complexity of off-nominal threads	Many or very complex off-nominal threads
Requirements straight forward	Some requirements complex	Requirements are complex
Very few COI's resulting	Average number of COI's	Many COI's resulting from scenario

3. Number of Critical Operational Issues (COI's)

Defining the COI's is another step in the test planning process. COI's are usually in the form of broad questions about the usability of the system in various mission scenarios. They are usually specified by the users in collaboration with the test planners, and program managers. The number can be calculated by counting the number of questions associated with each mission scenario, or the subjects that reflect controversies and uncertainties, usually documented in the test reports and test objectives documents.

Table 6: Number of Critical Operational Issues Definition

<u>*Number of Critical Operational Issues*</u> *COIs are key operational effectiveness or suitability issues expressed in the form of questions that reflect controversies and uncertainties about system capabilities, practicability, environmental effects, etc. COIs are examined in tests during the solution implementation phase to determine the SoS's capability to perform its mission.*

Table 7: Number of Critical Operational Issues Rating Scale

Easy	Nominal	Difficult
Clearly defined	Loosely defined	Badly defined
Easy to identify	Some can be identified	Difficult to identify
Resources easily found to support addressing the issue	Resources can be found to support addressing the issue	Difficult to find resources to support addressing the issue
Has many measures supporting its validity	Has adequate measures supporting its validity	Does not have adequate measures supporting its validity

4. Number of Measures of Performance, Effectiveness and Suitability

Measures of Performance (MOPs), Effectiveness (MOEs) and Suitability (MOSs) can be represented by single dimensional units like hours, meters, nanoseconds, dollars, number of reports, number of errors, number of CPR-certified employees, length of time to design hardware, etc. They are quantitative measures that are assigned to each COI during the test planning phase.

Table 8: Number of Measures of Performance, Effectiveness and Suitability Definition

<u>*Number of Measures of Performance, Effectiveness and Suitability*</u>
Measures of Effectiveness (MOEs) are quantitative measures that give some insight into how effectively a unit is performing. In addition, beyond the ability of the systems to support the functionality and performance called for by the SoS, there can be differences among the systems in characteristics that contribute to SoS "suitability" (MOSs) such as reliability, supportability, maintainability, assurance, and safety. Measures are assigned to each COI during the test planning phase of the test process. This driver seeks to capture the total number of measures assigned to COI's since these would all represent potential test points.

Table 9: Number of Measures of Performance, Effectiveness and Suitability Rating Scale

Easy	Nominal	Difficult
Clearly defined	Loosely defined	Badly defined
Easy to identify	Some can be identified	Difficult to identify
Resources can be found to support the measure	Resources can be found to support the measure	Difficult to find resources to support the measure
Already exists and used frequently in the past	Already exist and has been used	Does not already exist
Traceable to source	Can be traced to source with some effort	Hard to trace to source
High degree of overlap	Some overlap	Little overlap

5. Number of Systems in the SoS

The number of systems is a very important size driver because it defines how many systems need to be coordinated, which had a direct impact on the size of the effort. This number can typically be quantified by counting the individual systems used for testing as well as those in the SoS being tested, either physically, from documents or the blocks on a flow diagram showing the test procedures.

Table 10: Number of Systems in the SoS Definition

<u>*Number of Systems in the SoS*</u>
This driver represents the number of systems being tested within the SoS framework. This quantity is inclusive of individual components from various service branches, communication and networking systems, and all support equipment needed to test the systems.

Table 11: Number of Systems in the SoS Rating Scale

Easy	Nominal	Difficult
All used before	Mostly familiar, few not	Mostly new systems
Cohesive	Moderate cohesion	Low cohesion
Well behaved	Predictable behavior	Poorly behaved
All familiar requirements	Mostly familiar requirements	All new requirements
Low autonomy level	Average autonomy level	High autonomy level

6. Number of SoS Interfaces

System interfaces are also important drivers of UASoS T&E because both the quantity and complexity of interfaces comes at a price and requires more effort to ensure complete T&E. These interfaces typically can be quantified by counting the number of external and internal system interfaces among the SoS elements and from interface control documentation. However care needs to be taken to ensure that there is only focus on the technical interfaces, only count those interfaces that relate to the T&E process, determine the number of unique interface types and know the distinction between the SoS interfaces and the T&E interfaces, understand clearly the complexity of the interfaces as this plays into the interface ratings.

Table 12: Number of SoS Interfaces Definition

Number of SoS Interfaces

This driver represents the number of shared physical and logical boundaries between SoS components or functions (internal interfaces) and those external to the SoS (external interfaces) and particularly interfacing with testing equipment. For simplicity, please consider those interfaces between constituent systems.

Table 13: Number of SoS Interfaces Rating Scale

Easy	Nominal	Difficult
Simple message	Moderate complexity	Complex protocol(s)
Uncoupled	Loosely coupled	Highly coupled
Cohesive	Moderate cohesion	Low cohesion
Well behaved	Predictable behavior	Unpredictable behavior
Only one domain represented	Two or Three domains represented	All five domains represented

7. *Number of Tests*

The number of tests is directly related to the MOPs, MOEs and MOSs specified during the test planning phase. It can typically be quantified by counting the number of tests outlined in the test plans, or physically counting the number of test points actually conducted during the test mission as indicated in the evaluation reports. There needs to be a clear distinction between tests and retests. Retests are not accounted for in this driver. The number of distinct tests that are specified in the documentation is counted, but various smaller tasks within that major task are not to be included.

Table 14: Number of Tests Definition

Number of Tests
This driver represents the number of tests that have been identified to be conducted for ensuring the completion of the SoS testing and ensuring that it is ready for deployment. This includes a series of tests within a larger testing effort to make the SoS ready for various operational scenarios.

Table 15: Number of Tests Rating Scale

Easy	Nominal	Difficult
Clearly defined	Loosely defined	Badly defined
Easy to identify	Some can be identified	Difficult to identify
Timelines not an issue	Timelines is a constraint	Tight timelines
Requirements straight forward	Some requirements complex	Requirements are complex
Low risk	Medium risk	High risk

8. *Number of Stakeholders Involved*

The number of stakeholders can typically be quantified by physically counting the number of people on the test ranges, the test planners involved in laying out the test plans, the contractor/owners/organizations for the development of the various systems. They include program managers, program executive officers, contractors, users, engineers, and testers both at the system level and the SoS level. These numbers can be very different from one project to the next and again it is important to draw the appropriate boundaries to only get those directly involved the T&E process.

Table 16: Number of Stakeholders Involved Definition

Number of Stakeholders Involved *This driver represents the number of stakeholders who are involved in the test process. These include owners of the individual systems, contractors, oversight/integrators, testers, test engineers, test planners, as well as those responsible for the overall SoS project. All of those persons who have some stake in the test effort need to be accounted for.*

Table 17: Number of Stakeholders Involved Rating Scale

Easy	Nominal	Difficult
Clearly defined	Loosely defined	Badly defined
Easy to identify	Some can be identified	Difficult to identify
Communication is great	Communication is somewhat strained	Communication is terrible
Aware of each other	Only somewhat aware of each other	Not aware of each other
Have vested interest in the overall SoS	Have some interest in overall SoS	Have no interest in the overall SoS

Cost Drivers

The cost drivers in the model represent the multiplicative part of the model. They are called the effort multipliers since they affect the entire UASoS T&E effort calculation in a multiplicative manner. Assigning ratings for these drivers is not as straight forward as the size drivers because most of the cost drivers are qualitative in nature and require subjective assessment in order to be rated. In the COCOMO II model, a group of drivers were developed and used to reflect product, platform, personnel, and project factors that have been shown to influence cost and schedule for software projects (B. W. Boehm et al., 2000). COSYSMO recognized a number of themes that were not reflected in COCOMO II including understanding, complexity, operations, people and environment (Valerdi, 2005). COSOSIMO built on the COSYSMO themes by grouping them into categories and showing how each of these categories addresses the SoS Engineering (SoSE) core elements, as show in Table 18 below (J. A. Lane, 2009).

Table 18: Mapping of DoD SoSE Core Elements to COSYSMO Parameters

COSYSMO Parameters	SoSE Core Element
Requirements understanding	Translating capability
Architecture understanding Migration complexity Technology risk Number of recursive levels in the design	Understanding systems and relationships
Level of service requirements	Assessing actual performance to capability objectives
Architecture understanding Multisite coordination	Developing, evolving, and maintaining an SoS architecture/design
Level of service requirements Multisite coordination	Monitoring and assessing changes
Requirements understanding Architecture understanding Migration complexity Technology risk	Addressing new requirements and options
Personnel/team capability Personnel experience/continuity Process capability Multisite coordination Tool support	Orchestrating upgrades to SoS Stakeholder team cohesion

The three approaches described were either not appropriate for UASoS T&E effort, or were inadequate in addressing all the potential "causes" of cost. Therefore, using the Boehm's methodology in COCOMO II, COSYSMO drivers and the adaptation in COSOSIMO, this model seeks to create appropriate themes and cost drivers that address the major risks of UASoS T&E. Dahmann et al detailed some of these risks in their paper *"Systems of Systems Test and Evaluation Challenges"* (Dahmann, Rebovich, J. A. Lane, & Lowry, 2010). Their research as well as other documentation and discussions with stakeholders were used to create an initial list of potential cost drivers for UASoS T&E, and this was evaluated using the inputs of subject matter experts.

Initial evaluation of the potential cost drivers in UASoS T&E

The opinions of experts involved in the T&E of UASoS on the initial technical and organizational cost drivers initially identified as inputs to the cost model were collected (Deonandan, Valerdi, & J. A. Lane, 2010). Everyone interviewed or solicited ideas from had been involved in the T&E process for at least 10 years, either as a tester, test engineer, test planner, evaluator or program manager. 10 respondents completed the survey. They were asked to rate the identified risks on a scale of 1 to 5, with 5 having the greatest impact on the effort required for UASoS T&E and 1 having the smallest impact. These inputs were gathered to help prioritize the cost drivers, which are a combination of factors affecting SoS, individual systems and the testing process. This process was also used as a means to gather feedback on what drivers need to be changed, reworded, eliminated or added.

The following charts represent the inputs of subject matter experts in the area UASoS T&E. A score that is 3.5 and above represents a high impact driver, 2.5 to 3.49 represents a driver of medium impact and a driver with a rank below 2.5 is a low impact driver. **Error! eference source not found.** shows the responses to the technical drivers presented to respondents and the average score rating for each driver. The cost drivers rated higher were considered for further development. These results confirm the initial hypothesis that the T&E community prioritizes tests based on how complex the task is. Number of systems, integration complexity, number of requirements, technology maturity, synchronization complexity, requirements changes test complexity and diversity are all rated very high in their impacts on effort for SoS testing. Power availability was rated with least impact and conversations with respondents confirm that power issues can be easily remedied as opposed to the other factors that need to be considered. Additional cost drivers identified include emergent behaviors, data analysis tool capabilities and instrumentation requirements and changes.

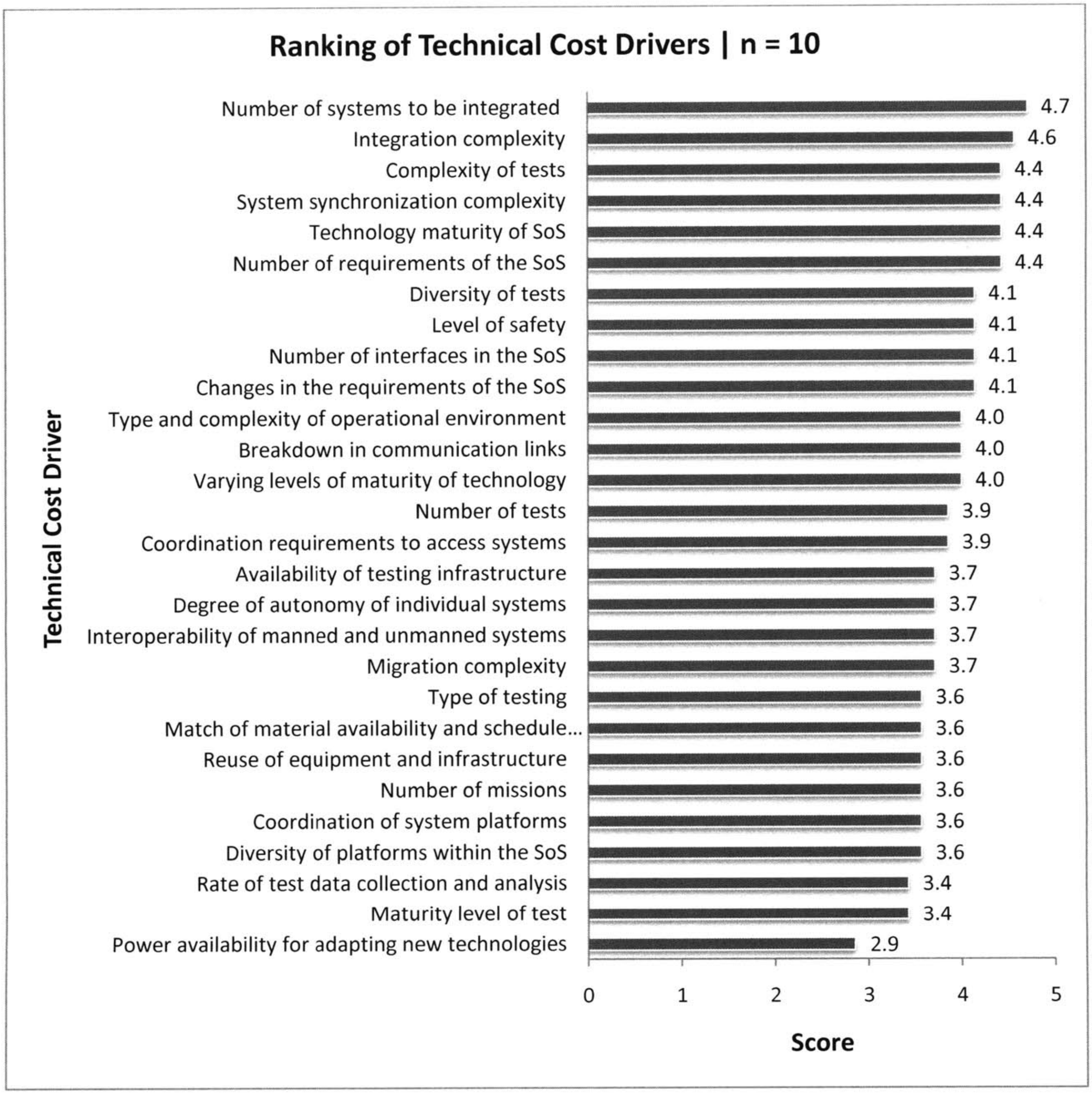


Figure 2: Initial Ranking of Technical Cost Drivers

Figure 3 shows the responses to the organizational drivers presented to respondents and the average score rating for each driver. From the organizational perspective it can be seen that understanding of the SoS requirements and architecture as well as the personnel availability and capability are rated as higher cost drivers compared to multisite coordination or stakeholder team cohesion. "Time constraints" is the most significant organizational driver of cost in T&E of UASoS.

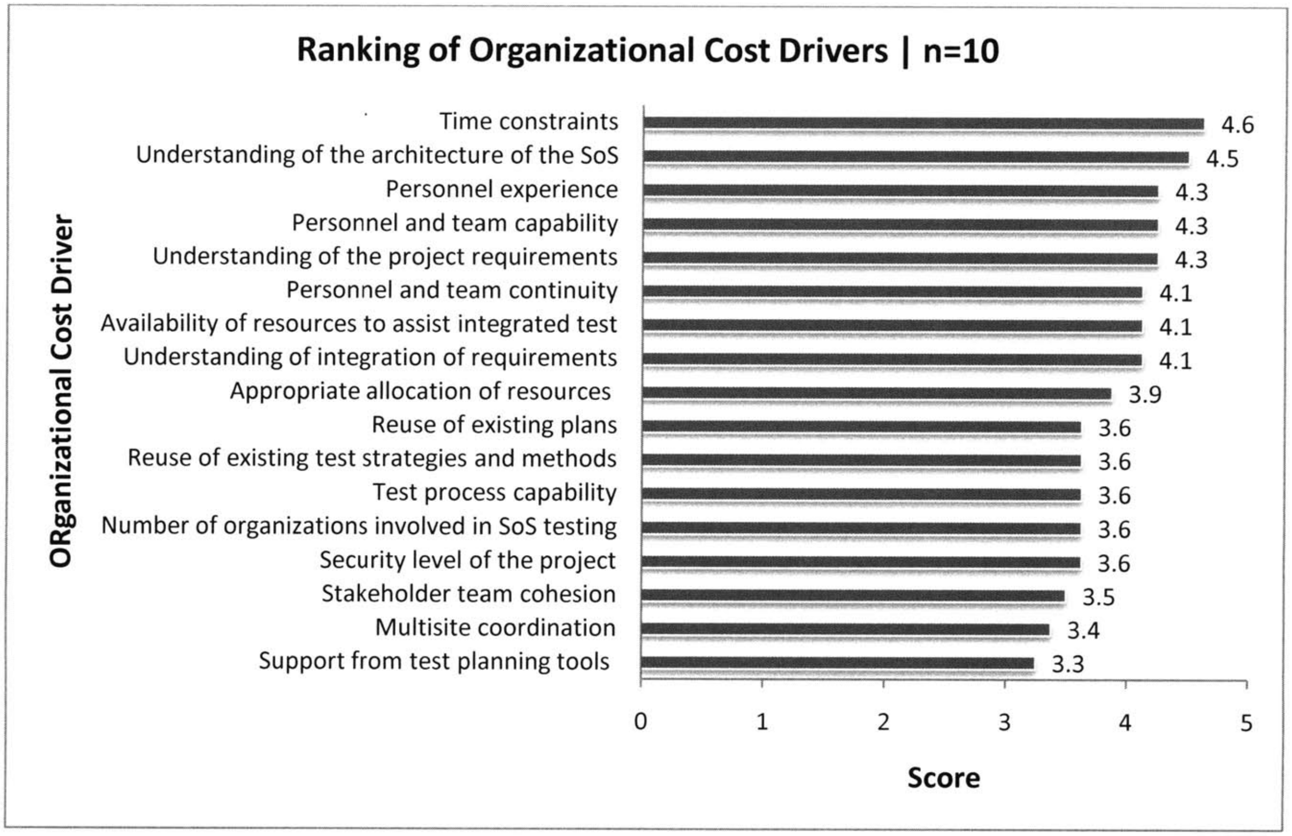


Figure 3: Initial ranking of organizational cost drivers

Model Cost Drivers

Those parameters rated medium or high impact were considered in the development of the cost drivers used in this model. The final list of cost drivers is shown in Table 19 . Many of these parameters are similar to those of COSYSMO, but the definitions, provided later in this section, have been modified to accommodate the unmanned, system of system and testing characteristics of this model.

Table 19: UASoS T&E Cost Estimation Model Themes and Cost Drivers

Theme	Parameters/Cost Drivers
Complexity	**Migration complexity** Legacy contractor Effect of legacy system on new system **System synchronization/integration complexity** Synchronization: Life Cycle Stage Integration: Technology Maturity **Technology Risk of SoS Components** Lack of Maturity Lack of Readiness Obsolescence **Level of Autonomy** **Test Complexity Level** Test Maturity Test Type Test Sensitivity
Operations	**Interoperability of manned and unmanned systems** **Flexibility** Technical Adaptability Program Adaptability **Synchronization of installations/platforms/tests in the SoS domain** Sites/Installations Operating Environment **Unexpected and undesirable emergent behavior** **Rate of test data collection and analysis** Frequency Adaptability **Level of automation in data collection integration and analysis** **Documentation match to testing needs** Formality Detail
Understanding	**Integration Requirements understanding** **Architecture Understanding**
People	**Stakeholder team cohesion** Culture Compatibility Familiarity **Personnel experience/continuity** Experience Annual Turnover **Personnel/team capability** **Test Process capability**
Test Environment	**Schedule Constraints** Test Planning Test execution and analysis **Testing Resource Challenges** Availability Allocation

The themes used here are similar to that of COSYSMO and are defined as follows:

- Complexity: Drivers that capture the difficulty, risk, and program-related factors that can influence UASoS T&E
- Operations. Drivers that capture how tests are conducted for UASoS T&E and the ability of the program to adapt to changes
- Understanding. Drivers that capture the level of comprehension and familiarity of the UASoS T&E team particularly when dealing with requirements and architecture
- People. Drivers that capture the capability of the UASoS team
- Test Environment. Drivers that capture the level of sophistication under which UASoS T&E is performed

Each driver was also assigned a rating scale that described different attributes that could be used to rate the degree of impact on the T&E effort. These can be thought of as knobs that can be turned to different levels depending on the impact on cost. The rating levels included: Very Low, Low, Nominal, High, Very High, and Extra High. The Nominal level represents zero impact and is assigned a multiplier of 1.0. Levels above and below nominal are assigned multipliers above or below 1.0 to reflect their impact on systems engineering effort. The increase or decrease of multipliers along the rating scale depends on the polarity of each driver.

Complexity Factors

Complexity factors account for variation in effort required to in UASoS T&E caused by the characteristics of the test process and the individual systems within the UASoS. When efforts have to be conducted on very different systems at once, immature technologies, systems with high degrees of autonomy, and complex testing procedures, they will take longer to complete. 5 drivers in this model are associated with complexity, including: *Migration Complexity, System*

Synchronization/Integration Complexity, Technology Risk of SoS Components, Level of Autonomy, and Test Complexity.

1. ***Migration Complexity:*** This driver takes into consideration how UASoS are currently tested in the field. Many traditional systems are integrated with newer systems and this driver rates the extent to which these legacy systems influence the test effort. It is divided into two main fields: The effect of the legacy contractors and the effect of the legacy systems themselves on the test effort. In the first case, the nominal situation would be if the contractors, testers and developers of the systems are the same and all documentation is available to test all systems well. If all contractors, testers and developers are different and no documentation is available, this increases the cost of testing significantly. In the second case, if everything is new and no previous systems existed, costs would be expected to be nominal since the effort has never been done, but if newer systems need to be integrated with legacy systems, compatibility issues is expected to drive up the costs of testing. These ratings are described in the tables below.

Table 20: Migration Complexity Definition

Migration Complexity
This cost driver rates the extent to which the legacy SoS affects the migration complexity, if any. Legacy SoS components, databases, workflows, environments, etc., may affect the SoS test due to new technology introductions, planned upgrades, increased performance, etc.

Table 21: Migration Complexity Rating Scale

	Nominal	High	Very High	Extra High
Legacy Contractor	Self; legacy systems in SoS test are well documented. Original team largely available	Self; original development and test team not available; most documentation available	Different developers and testers; limited documentation	Original developers and testers no longer involved; no documentation available
Effect of legacy system on new system	Everything is new. No previous systems existed	Migration is restricted to integration test only	Migration is related to integration test and development test	Migration integration test, development test and requires more systems to be added for compatibility

2. System Synchronization/Integration Complexity: SoSs are expected to have long life cycles, from years to decades. When the systems are integrated, they may be at varying levels of maturity, and the way in which they interact with each other at one point in the program could be significantly different from other points as the SoS matures. Individual systems within the SoS may be at varying stages in their life cycles making synchronization difficult. Further, the individual systems within a SoS may have varying levels of maturity and may enter the SoS at different stages of the SoS lifecycle. Ensuring that these systems can still work together and merging newer more advanced technologies with more traditional technologies can present a significant challenge to development and validation of the SoS. Emergent risks and unanticipated program or technical problems may develop and the wider the difference in these systems, the more the effect on cost for testing as compatibility becomes an issue.

Table 22: System Synchronization/Integration Complexity Definition

System Synchronization/Integration Complexity *This cost driver rates the extent to which there is difficulty in adopting systems, whose procurement are not synchronized, at different stages of the life cycle of the SoS. This can include such examples as merging 50 year old technology with cutting edge technology*

Table 23: System Synchronization/Integration Complexity Rating Scale

	Very Low	Low	Nominal	High	Very High
Synchronization: Life Cycle Stage	All systems are at the same stage of their life cycle	Systems are at similar stages in their life cycle	Few systems are at different stages of the life cycle	Many systems are at different stages in their life cycle	Systems are at vastly different stages in their life cycle
Integration: Technology Maturity	All systems are at the same level of maturity so compatibility is no issue	Technology maturity is different but still adequate for full compatibility	Technology maturity is lacking in compatibility but few changes need to be made to help with some compatibility issues	Technology maturity is lacking in compatibility but many changes must be made to help with some compatibility issues	Technology at very different maturity levels and it is impossible for the systems to be compatible with each other

*3. **Technology Risk of SoS Components:*** On a SoS level, the difficulty in integration can be underestimated, while the maturity of the individual systems is overestimated creating a technology risk in integration. This is compounded by an increase in the number of installations and platforms to be dealt with as well as the migration complexity. While these problems may not manifest themselves at the beginning, as the SoS tries to become more integrated and developed, these disconnects among the components will surface resulting in costs, stagnation in growth, and loss in performance of the SoS. This driver accounts for technology risks in three ways. It measures the general maturity levels of the component systems, how ready the technology is for integration using the Technology Readiness Levels (TRL's), and how obsolete the technology is. The lower the maturity level, the technology readiness level, and the more obsolete the technology is, the higher the costs expected.

Table 24: Technology Risk of SoS Components Definition

Technology Risk of SoS Components *The maturity, readiness, and obsolescence of the technology being implemented. Immature or obsolescent technology will require more testing effort.*

Table 25: Technology Risk of SoS Components Rating Scale

	Very Low	Low	Nominal	High	Very High
Lack of Maturity	All technology proven and already implemented in other areas	Most technology proven through actual use and ready for widespread adoption	All technology is proven on pilot projects and ready to roll-out for production	Most systems are ready for pilot use	Most systems still in the laboratory stages. All systems are at the same stage of their life cycle
Lack of Readiness	Mission proven (TRL 9)	Concept qualified (TRL 8)	Concept has been demonstrated (TRL 7)	Proof of concept validated (TRL 5 & 6)	Concept defined (TRL 3 & 4)
Obsolescence	----	----	Technology is the state-of-the-practice; Emerging technology could compete in future	Technology is stale; New and better technology is on the horizon in the near-term	Technology is outdated and use should be avoided in new SoS; Spare parts supply is scarce

*4. **Level of Autonomy:*** Autonomy and advances in technologies have facilitated the expansion of capabilities without the need to endanger human life; however, substantial costs are required to fund such projects, not to mention the losses incurred if the control mechanisms fail or the system is lost. Future DoD systems will have autonomous aspects where the systems will be unmanned and able to be self aware and recognize the physical environment in which they operate. However, because these systems are so complex, the interconnected parts have more properties and control and operator interfaces can be drastically different especially with the variation in autonomy levels from one system to the next. It is also important to note that the degree of autonomy of the individual systems can result in cost savings in some areas and additional costs in other areas. From an operational perspective, it may be less costly to operate a set of systems with a higher degree of autonomy because the systems are more developed and capable of fulfilling the missions while maintaining safety to the warfighter. From the development perspective, the higher the degree of autonomy, the more significant the costs especially when ensuring that the UAS meets its specified requirements and is capable of maintaining the safety of the warfighter. This driver rates the level of autonomy of the systems within the UASoS, with the assumption that the system with the highest level of autonomy within the UASoS would be the determining rating for the driver, and that the higher the level of human independence, the higher the costs will be.

Table 26: Level of Autonomy Definition

Level of Autonomy *This cost driver rates the level of autonomy of individual systems that creates need for more planning and coordination*

Table 27: Level of Autonomy Rating Scale

Very Low	Low	Nominal	High	Very High
No human independence, lowest mission complexity, lowest environmental complexity	Low level of human independence, low level tasks and simple environments	Mid level human independence, mid complexity, multi functional missions, moderate environments	High level human independence, collaborative, high complexity missions, difficult environments	Approaching 100% human independence, highest complexity, all missions in extreme environments

5. Test Complexity Level: When tests are implemented, the degree of risk is directly related to the impact of a possible failure of the system. This cost driver rates how severely the complexity of a test impacts SoS testing capabilities. It is divided into three main categories, focusing on test maturity, test type and test sensitivity. When a large number of tests need to be performed on a large number of systems this can be very complex. The type of test and amount of each type of test to be performed will be drivers of costs. For example, field tests require considerable resources, labor, and scheduling, and are significantly more costly than a simulated test which can be done in a virtual environment. Testing systems in specific domains can also be difficult especially in the space and undersea arenas which are primarily UAS environments and access becomes exponentially more difficult and expensive. The maturity level of the test which defines how evolved the test and test process are, can influence the ability of a test to predict whether the SoS has met its expected requirements and capabilities. The more evolved and established the test procedure is, the lower the costs will be. Further, the impacts of test failures can range from annoyance to total system crash, with more cost incurred as the degree of impact gets worse. The tables below document the ratings for the test complexity cost driver.

Table 28: Test Complexity Level Definition

Test Complexity Level *When tests are implemented, the degree of risk is directly related to the impact of a possible failure of the system. This cost driver rates how severely the complexity of a test impacts SoS testing capabilities.*

Table 29: Test Complexity Level Rating Scale

	Nominal	High	Very High	Extra High
Test Maturity	Tests have been done before, and are all similar to each other.	Most tests have been done in the past and are not very complex	Some test have been done in the past, but others are very complicated to do	Tests have never been done before, are extremely diverse and complicated to do
Test Type	Only hardware and software tests	Hardware and software test in a live environment	Hardware and software test in a live and virtual environment	Hardware, software, live, virtual and constructive tests
Test Sensitivity	The failure causes inconvenience or annoyance (e.g., cosmetic errors, awkward navigation)	The failure causes impairment of critical or essential system functions, but a workaround solution does exist	The failure causes impairment of critical or essential system functions and no workaround solution exists	The failure causes a system crash, unrecoverable data loss, or jeopardizes human safety

Operations Factors

The operations factors refer to the hardware and software environments that a system will operate within, the interactions between the systems and the processes during operation and how the environment impacts these interactions. Depending on the UASoS of interest, the operational domains can be space, air, land, sea, and undersea, and the platform can be an aircraft carrier; an aircraft; an airborne missile; a navigation, guidance, and control system; or a level of the computer systems software infrastructure. The existence of legacy issues may also impact the amount of effort required to incorporate the new system with existing technologies and cultures for effective operation. In this model, 7 operations cost drivers have been identified. These include: *Interoperability of manned and unmanned systems, Flexibility, Synchronization of installations/platforms/tests in the SoS domain, Unexpected and undesirable emergent behavior, Rate of test data collection and analysis, Level of automation in data collection integration and analysis, Documentation match to testing needs.*

***6. Interoperability of Manned and Unmanned Systems*:** UASoS function and operate in open, non-deterministic environments and are more focused on interactions between components, both manned and unmanned. The interconnected parts have more properties, and control and operator interfaces can be drastically different. Therefore, a UASoS requires the ability for manned and unmanned systems to co-operate with each other to fulfill its purpose. However the degree of safety of these interactions is limited by the protective barriers put in place and especially during the verification stage when many of these risks are still unknown and unpredictable, the ability for both manned and unmanned systems to co-operate depends on the outcomes of these tests. Some of the higher level risks of UASoS include unintended or abnormal system mobility operation, inadvertent firing or release of weapons, engagement or firing upon unintended targets, self-damage of own system from weapon fire or release, personnel injury, equipment damage, environmental damage, system loss and system collision. The greater these risks the more costly the test effort will be to help identify and minimize these risks.

Table 30: Interoperability of Manned and Unmanned Systems Definition

<u>Interoperability of Manned and Unmanned Systems</u> *This cost driver rates the level of complexity of integrating both manned and unmanned systems into the SoS. This looks at the level of communication and coordination that can be expected from the systems on the SoS and how this impacts the complexity of the test.*

Table 31: Interoperability of Manned and Unmanned Systems Rating Scale

Very Low	Low	Nominal	High	Very High
SoS successfully demonstrates that it meets requirements and capabilities and all measures of effectiveness are clearly defined	SoS successfully demonstrates that it meets requirements and capabilities but all measures of effectiveness not clearly defined	SoS successfully demonstrates that it meets requirements but capabilities are not met and all measures of effectiveness not clearly defined	SoS successfully demonstrates that it has capabilities, but all requirements are not met and all measures of effectiveness not clearly defined	SoS does not successfully demonstrate that it meets requirements, capabilities are not met and all measures of effectiveness not clearly defined

*7. **Flexibility:*** UASoS offer the flexibility for additional capabilities, which manned systems or SoS are not capable of due to combined safety and effectiveness considerations and because UASoS must have the capacity for adaptation and change and be able to perform the unexpected no matter what mission it has to perform, there is need for the test infrastructure to adapt to changes throughout the test process. The number of requirements of the UASoS is a key driver of risk, as well as changes in requirements throughout SoS development and operation. But, not only do requirements change within a mission setting; missions and operational platforms also change resulting in changing requirements to reflect the warfighter's needs. From the technical aspect, this cost driver rates how the systems in the UASoS and test infrastructure themselves can be technically adapted to ensure the test effort continues despite changes in expectations. From the programmatic angle, it looks at how changes in the requirements, schedule or budget may cause the program to fail even if the technical capabilities are there. If flexibility is available throughout this would drive up costs as the program needs to recover each time there is a change. The ratings for this driver are described in more detail in the following pair of tables.

Table 32: Flexibility Definition

Flexibility *This cost driver rates ability of the test effort to adapt to technical and programmatic changes during the test program. From the technical aspect, it rates how the systems in the SoS and test infrastructure themselves can be technically adapted to ensure the test effort continues. From the programmatic angle, it looks at how changes in the requirements, schedule or budget may cause the program to fail depending even if the technical capabilities are there. This cost driver rates how the presence of emergent behaviors affects the testing of SoS when different systems are merged into the SoS*

Table 33: Flexibility Rating Scale

	Very Low	Low	Nominal	High	Very High
Technical Adaptability	Flexibility not present, cannot be added or configured to new systems	Flexibility can be added, but require too much configuration to be feasible for new system addition	Flexibility can be added and require configuring to adapt to the new systems	Flexibility available but require configuring to adapt to new systems	Flexibility available from beginning and easy to adapt to new systems
Program Adaptability	Flexibility not present, test effort ends because program cannot recover from changes	Flexibility can be added, but require too much reconfiguration to be feasible for program to recover from changes	Flexibility can be added and require reconfiguration for program to recover from changes	Flexibility available but require reconfiguration for program to recover from changes	Flexibility available from beginning and easy for program to recover from changes

8. ***Synchronization of installations/platforms/tests in the SoS domain*****:** A typical SoS integrates a number of operational platforms, a versatile mix of mobile, networked systems that will leverage mobility, protection, information and precision. To conduct effective operations across such a spectrum requires careful planning and co-ordination of space, air, land domain platforms and networks, understanding of the SoS architecture and capabilities, as well as interoperability across all components of SoS. This driver rates the synchronization in two ways. It looks at the number of physical sites/installations that need to be used to conduct the test, with the greater the diversity and the number of sites needed, the more costly the test. It also considers the operating environments, in terms of the domains and whether they are conducive to completing the tests. The harsher the operational environment, the more costly the test.

Table 34: Synchronization of Installations/Platforms/Tests in the SoS Domain Definition

Synchronization of installations/platforms/tests in the SoS domain *The synchronization of different platforms, that is installation sites, as well as the complexity of the operating environment that the SoS will entail, namely space, air, land, sea and undersea. It also looks at the ability to adequately combine different tests into one test plan to test the overall SoS requirements and capabilities, as well as the type of test that has to be performed.*

Table 35: Synchronization of Installations/Platforms/Tests in the SoS Domain Rating Scale

	Nominal	High	Very High	Extra High
Sites/Installations	Single installation site or configuration	2-3 sites or diverse installation configurations	4-5 sites or diverse installation configurations	>6 sites or diverse installation configurations
Operating Environment	Existing facility meets all known environmental operating requirements	Moderate environmental constraints; controlled environment	Ruggedized mobile land-based requirements; some information security requirements. Coordination between 1 or 2 regulatory or cross functional agencies required.	Harsh environment (space, sea, undersea airborne) sensitive information security requirements. Coordination between 3 or more regulatory or cross functional agencies required.

9. Unexpected and undesirable emergent behavior: One of the risks associated with UASoS is the occurrence of unexpected emergent behavior which occurs when systems are integrated for the first time. Since the systems are built separately by various contractors, and are usually brought together for operational testing, these behaviors would most likely be seen for the first time. It is wise to anticipate some of the problems that may arise beforehand. For example, architecture-level design and technology problems may show up in early to mid development, while manufacturing and integration problems may be present in mid to later development, and support related problems may follow system deployment. Further adding to this, some of them may not even manifest initially but as systems are put in various configurations, operational scenarios and environments, these behaviors can be teased out. The more of these there are the more tests are performed and the higher the costs.

Table 36: Unexpected and Undesirable Emergent Behavior Definition

Unexpected and undesirable emergent behavior *This cost driver rates how the presence of emergent behaviors affects the testing of SoS when different systems are merged into the SoS*

Table 37: Unexpected and Undesirable Emergent Behavior Rating Scale

Nominal	High	Very High	Extremely High
Minimal unexpected and undesirable emergent behaviors	Many unexpected and undesirable emergent behaviors	Many unexpected and undesirable emergent behaviors	Too many frequent unexpected and undesirable emergent behaviors

10. Rate of test data collection and analysis: This driver rates how often data is collected and analyzed and how adaptable the data collection process is to changes in T&E needs. It assumes that the less frequent the data collection and analysis, the less costly the effort would be. In terms of adaptability, it assumes that the more flexibility that is built into the system, the greater the likelihood that data collection and analysis can continue so that the T&E effort can be completed. The more changes that have to be made for the program to continue, that is the more there is flexibility built in, the more effort would be required.

Table 38: Rate of Test Data Collection and Analysis Definition

Rate of test data collection and analysis *This cost driver rates the efficiency in collecting and analyzing data while testing*

Table 39: Rate of Test Data Collection and Analysis Rating Scale

	Very Low	Low	Nominal	High	Very High
Frequency	Very little automation. Data analyzed at the end of test loop	Some automation in the collection but not the analysis. Data analyzed at the end of the test loop	Collected when needed and analyzed at the end of the test loop	High degree of automation in the collection. Data analyzed manually in real time	Very high degree of automation in both the collection and analysis of data. Data analyzed in real time
Adaptability	Flexibility not present, test effort ends because program cannot recover from changes	Flexibility can be added, but require too much reconfiguration to be feasible for program to recover from changes	Flexibility can be added and require reconfiguration for program to recover from changes	Flexibility available but require reconfiguration for program to recover from changes	Flexibility available from beginning and easy for program to recover from changes

11. Level of automation in data collection integration and analysis: As UASoS become more complex and more systems and interfaces are added, keeping track of these interactions becomes challenging. Individually and physically monitoring each interaction especially when multiple testing sites and domains may be involved becomes problematic. Having automated data collection and analysis capabilities makes this task less problematic, but come at a cost. This driver seeks to capture how sophisticated these technologies are and during the test process.

Table 40: Level of Automation in Data Collection Integration and Analysis Definition

Level of automation in data collection integration and analysis *Coverage, integration, and maturity of the automated tools used in test data collection and analysis*

Table 41: Level of Automation in Data Collection Integration and Analysis Rating Scale

Low	Nominal	High	Extra High
Single stand alone systems with minimal automation	Basic automated tools moderately integrated throughout the testing process	Strong, mature automated tools, moderately integrated with other disciplines	Strong, mature proactive use of automated tools integrated with process, model-based testing and management systems

12. Documentation match to testing needs: During operation it is important to have clear reporting, detailed enough to allow all bases to be covered but not with extraneous information that makes the team waste time. Therefore this driver looks at the match between what documentation requirements and the current testing needs. Because many of these UASoS are fielded for the first time, the requirements of this documentation may not be fully known, but there needs to be some basic guidance that the team could follow to make reports. This driver looks at the formality of the documentation, whether they are just generalized goals, or whether there are rigorous standards that are established to be followed in reporting. It also looks at the level of detail present in reporting requirements. The more detailed they are and the more revisions that need to make the greater the effort needed.

Table 42: Documentation Match to Testing Needs Definition

Documentation Match to Testing Needs *The formality and detail of documentation required to be formally delivered based on the testing needs of the system.*

Table 43: Documentation Match to Testing Needs Rating Scale

	Very Low	Low	Nominal	High	Very High
Formality	General goals, stories	Broad guidance, flexibility is allowed	Risk-driven degree of formality	Partially streamlined process, largely standards-driven	Rigorous, follows strict standards and requirements
Detail	Minimal or no specified documentation and review requirements relative to life cycle needs	Relaxed documentation and review requirements relative to life cycle needs	Risk-driven degree of formality, amount of documentation and reviews in sync and consistent with life cycle needs of the system	High amounts of documentation, more rigorous relative to life cycle needs, some revisions required	Extensive documentation and review requirements relative to life cycle needs, multiple revisions required

Understanding Factors

This cost driver theme deals with the UASoS T&E team's comprehension of and familiarity with the system of interest. Higher ratings for these drivers represent a productivity savings. There are two understanding factors in this model, including: *Integration requirements understanding*, and *Architecture understanding*.

13. Integration Requirements Understanding: Many times it is unclear what the SoS needs to do in order to fulfill its mission and without the appropriate metrics to evaluate the performance of the UASoS, it is difficult to determine whether the mission is successful or not. Further, while some stakeholders may provide high level requirements in the form of system capabilities, objectives or measures of effectives, some stakeholders may need to break these requirements down to help fully integrate these requirements and this will require a thorough understanding of the system. Counting the number of requirements and rating their complexities is addressed by the size driver. But the overall degree of understanding of these requirements – by all the stakeholders – has a multiplicative effect on the total amount of effort needed for systems engineering. The more requirements change, the greater the effort for testing.

Table 44: Integration Requirements Understanding Definition

Integration Requirements Understanding
This cost driver rates the level of understanding of the requirements for integration depending on the stage in the testing process. This includes the understanding by all stakeholders including the systems, software, hardware, customers, team members, users, and especially the testers etc. . Primary sources of added testing effort are unprecedented SoS, unfamiliar domains, or SoS whose requirements are emergent with use.

Table 45: Integration Requirements Understanding Rating Scale

Very Low	Low	Nominal	High	Very High
Full understanding of requirements, familiar system	Strong: few undefined areas	Reasonable: some undefined areas	Minimal: many undefined areas	Poor: emergent requirements or unprecedented system

14. Architecture Understanding: On a SoS level, it is essential to understand the architecture of the system, its associated infrastructure, and the interactions between each system within the system of systems. Understanding the architecture is also important in designing test processes. This is different from requirements understanding and therefore warrants its own driver. Other than unprecedentedness and domain unfamiliarity, primary sources of added effort are new technologies, complex COTS products and choices, varying levels of maturity in systems and interfaces, and depth of the Work Breakdown Structure (WBS). The higher the complexity of integrating a diverse set of systems and associated interactions creates a more risky environment as individual systems may be at various levels of maturity. Therefore, the lower level of understanding of the architecture, the more effort has to be put into the T&E effort.

Table 46: Architecture Understanding Definition

Architecture Understanding *This cost driver rates the relative difficulty of determining and managing the system architecture in terms of platforms, standards, components (COTS etc), connectors (protocols), and constraints. This includes tasks like systems analysis, tradeoff analysis, modeling, simulation, case studies, etc.*

Table 47: Architecture Understanding Rating Scale

Very Low	Low	Nominal	High	Very High
Full understanding of architecture and the connections and interoperability of constituent systems, all familiar	Strong understanding of architecture and the connections and interoperability of constituent systems, few unfamiliar areas	Reasonable understanding of architecture and the connections and interoperability of constituent systems, some unfamiliar areas	Minimal understanding of architecture and the connections and interoperability of constituent systems, many unfamiliar areas	Poor understanding of architecture and the connections and interoperability of constituent systems

People Factors

People factors have a strong influence in determining the amount of effort required to conduct UASoS T&E. These factors are for rating the T&E team's vs. the individual's capability and experience and for rating the project's process capability. There are four people factors considered in this model including: *Stakeholder team cohesion, Personnel/Team capability, Personnel experience/continuity, and Test process capability.*

15. Stakeholder team cohesion: The mutual culture, compatibility, familiarity, and trust of the stakeholders involved in the T&E effort are key factors that have significant importance in ensuring UASoS are tested sufficiently. Because a UASoS deals with so many different types of systems, it is important for stakeholders to think broadly about how the UASoS will deliver its capabilities without being caught up with how one system performs. The more diverse thinking there is, the less effort it will take to ensure everyone is working towards a common goal. There also needs to be strong collaborations, and clear roles and responsibilities with stakeholders. Absence of this can result in conflicting organizational objectives and increases costs.

Stakeholder familiarity with the processes as well as each other is also important, as working this helps to promote collaboration and minimize costs.

Table 48: Stakeholder Team Cohesion Definition

<u>*Stakeholder Team Cohesion*</u> *Represents a multi-attribute parameter which includes leadership, shared vision, and diversity of stakeholders, approval cycles, group dynamics, IPT framework, team dynamics, and amount of change in responsibilities. It further represents the heterogeneity in stakeholder community of the end users, customers, implementers, and development team.*

Table 49: Stakeholder Team Cohesion Rating Scale

	Very Low	Low	Nominal	High	Very High
Culture	Stakeholders with diverse domain experience, task nature, language, culture, infrastructure. Highly heterogeneous stakeholder communities	Heterogeneous stakeholder community. Some similarities in language and culture	Shared project culture	Strong team cohesion and project culture. Multiple similarities in language and expertise	Virtually homogeneous stakeholder communities. Institutionalized project culture
Compatibility	Strong mutual advantage to collaboration	Clear roles & responsibilities	Compatible organizational objectives	Converging organizational objectives	Highly conflicting organizational objectives
Familiarity	Extensive successful collaboration	High level of familiarity	Some familiarity	Willing to collaborate, little experience	Unfamiliar, never worked together

16. Personnel/Team Capability: This driver combines the intellectual horsepower of the team members, how much of the process horsepower is focused on the problems, and the extent to which the horsepower is pulling in compatible directions. It is measured with respect to an assumed national or global distribution of team capabilities (Valerdi, 2005).

Table 50: Personnel/Team Capability Definition

<u>*Personnel/Team Capability*</u> *Basic intellectual capability of a SoS testing team (compared to the overall testers of SoS) to analyze complex problems and synthesize solutions.*

Table 51: Personnel/Team Capability Rating Scale

Very Low	Low	Nominal	High	Very High
15th percentile	35th percentile	55th percentile	75th percentile	90th percentile

17. Personnel experience/continuity: This driver rates how experienced personnel are in a particular project. Many times, UASoS are being fielded for the first time and such combinations may never have existed in the past. However, the extent to which the same personnel can be used in testing UASoS the less the cost will be because they will bring in the experience from previous projects which may be applied in similar even though not identical circumstances. However, often times many years of experience does not translate to competency in a certain area. Experience is rated as of the beginning of the project and is expected to increase as the project goes on, unless adversely affected by personnel turnover. In addition, if turnover is high, then more costs will be incurred as personnel have to be retrained on testing procedures and expectations. Therefore, this driver is divided into two categories, Experience and Annual Turnover.

Table 52: Personnel Experience/Continuity Definition

<u>*Personnel Experience/Continuity*</u> *The applicability and consistency of the staff throughout the test project with respect to the domain, customer, user, technology, tools, etc.*

Table 53: Personnel Experience/Continuity Rating Scale

	Very Low	Low	Nominal	High	Very High
Experience	10 years of continuous experience	5 years of continuous experience	3 years of continuous experience	1 year continuous experience, other technical experience in similar job	Less than 2 months
Annual Turnover	48%	24%	12%	6%	3%

18. Test Process Capability: This driver rates how established the test process is and focuses on how personnel's mindset throughout the process. It looks at how focused personnel are on managing and optimizing the processes to adapt to changes, and the ability to strategize for improvements. The more defined the test process, the more measures that have been put in place to ensure seamless T&E, and the more optimized and flexible the test process becomes throughout the program will require effort. The direction that the test process could take is in itself very unpredictable because many UASoS have not existed in the past and it is next to impossible to predict what would be required as the process continues particularly with unexpected emergent behaviors. The more measures that are put in place to deal with these issues, the more effort would be required since personnel need to adapt to these processes. Therefore, this driver necessary to measure the consistency and effectiveness of the project team in performing the test process, and rates how the team is capable of adjusting to new demands in the test process.

Table 54: Test Process Capability Definition

Test Process Capability *The consistency and effectiveness of the project team at performing testing processes. This can also be based on project team behavioral characteristics, if no assessment has been performed.*

Table 55: Test Process Capability Rating Scale

Very Low	Low	Nominal	High	Very High	Extra High
Ad Hoc approach to test process performance	Performed test process, activities driven only by immediate user requirements, Test focus limited	Managed test process, activities driven by users needs in a suitable manner, Test focus is requirements through mission scenarios – not driven by organizational processes	Defined test process, activities driven by benefit to capabilities, Test focus is through mission scenarios, process approach driven by organizational processes tailored for the test program	Quantitatively managed test process, activities driven by test benefit, Test focus on both the developmental and operational environments	Optimizing test process, continuous improvement, activities driven by user and organizational benefit, Test focus is developmental and operational environments and strategic applications

Test Environment Factors

These drivers capture the level of sophistication under which UASoS T&E is performed, what the demands and support for UASoS T&E. There are two drivers associated with the test environment: *Schedule constraints, and Testing resource challenges.*

19. Schedule Constraints: This driver focuses on the time pressures that affect the test process. As requirements change, capabilities become more critical and priorities shift, this places additional pressures on the team to ensure that they meet schedule constraints. Being able to take the UASoS from initial testing to the point of delivery requires careful replanning in light of any changes in demands and most likely many changes in the test execution timeline. If times are extended then less emphasis would be needed on additional resources and personnel to complete the tasks whereas with schedule constraints, this can become more costly. This driver seeks to capture these changes in schedules.

Table 56: Schedule Constraints Definition

Schedule Constraints

This driver is a multi-attribute parameter that rates the time pressures that affect the testing process. It represents the amount of time that is necessary to ensure full testing of the entire SoS, as well as changes made to the schedule during the testing process. This includes adopting systems at various points in the life cycle, synchronizing systems at their individual points in their life cycles, basically taking the SoS through all testing to the point of delivery. Nominal here can be thought of as the predefined "adequate" amount of time that it takes to complete testing of the SoS.

Table 57: Schedule Constraints Rating Scale

	Very Low	Low	Nominal	High	Very High
Test Planning	160% of nominal	130% of nominal	100% of nominal	85% of nominal	75% of nominal
Test Execution and Analysis	160% of nominal	130% of nominal	100% of nominal	85% of nominal	75% of nominal

20. Testing Resource Challenges: This driver focuses specifically on the availability and allocation of resources for the test process and how much administrative overhead is required to ensure that resources are where they are needed at the right time. When most of the resources are not available at the right time this extends the project timeline and places additional pressures on program managers and testers to reschedule. This becomes even more problematic when dealing with systems from multiple domains in multiple sites, and ensuring that resources are allocated appropriately and coordinated when a test needs to be performed. During the test planning phase, it is essential to ensure that all resources will become available when needed and allocated appropriately in order to save time, money and labor. The more badly allocated and less available resources are the greater the effort needed to ensure tests are completed in time.

Table 58: Testing Resource Challenges Definition

<u>*Testing Resource Challenges*</u> *This driver is a multi-attribute parameter that rates the resource (testing infrastructure) challenges faced during the testing process. It represents the availability of resources and substitutes for these resources for the various phases of testing, as well as the amount of paperwork that needs to be done to ensure these resources can be used at the appropriate times. It also represents how well these resources have been allocated for testing.*

Table 59: Testing Resource Challenges Rating Scale

	Very Low	Low	Nominal	High	Very High
Availability	All resources available, substitutes available, and no paperwork required	All resources available, substitutes available, and paperwork requirements are minimal	Most resources available, substitutes are available and paperwork takes reasonable amount of time to complete	Some resources available, there are some substitutes and paperwork is complicated	Most resources not available, substitutes not available and paperwork takes too long to complete
Allocation	Allocation is very well done and there are even excess resources after testing completed	Resources well allocated with just enough for completion of testing	Resources reasonably well allocated but replanning has to be done to ensure completion with what is available	Resources badly allocated but replanning is able to take testing to completion	Resources badly allocated for the testing phases, and are exhausted before testing completion

Work Breakdown Structure

The work breakdown structure (WBS) presented here reflects the elements of the T&E process and relates it to the end product. By displaying and defining the tasks to be accomplished, the WBS becomes a management blueprint for a tested UASoS and can also be used to communicate management's plan for how a program is to be accomplished. The WBS also helps design the architecture for a project, establishes a baseline for reporting project status, and forms a basis for estimating the time and effort needed for the project. Prior to this effort, no standardized WBS for T&E existed across the services, so it was critical to develop one in order to understand the similarities and differences in how testing is done and get a good coverage of data across the services, especially since the individual systems in the UASoS could be from any service. In fact, much of the UASoS T&E is done jointly across the services. This WBS is not only used in the definition of the model, but also as a boundary around which data can be collected.

To create a WBS, two main methods can be used. The top-down approach begins with the project goal and keeps breaking down activities until the smallest task needed is accomplished, and the bottom-up approach establishes the top level activities using the top-down approach, and then breaking up these activities into sub categories. For UASoS T&E the bottom-up approach was used. There are four main activities involved in T&E. These include:

1. Test Planning
2. Test Readiness Review
3. Test Execution
4. Test Evaluation

They were determined based on analysis of the test documentation of the Army, Navy and Air Force, as well as interviews with test planners, testers and project managers within the DoD. Each of these categories was divided into sub categories shown in Table 60. The WBS was evaluated by cost estimators and DoD personnel based on seven criteria. These include:

1. Measurable Status – Is each task defined in to help monitor its status toward completion?
    - Typically requires some kind of measurement to assess percent completion
2. Bounded – Is each task clearly bounded by start and stop events?
    - What event marks the start and stop of each task?
3. Deliverable – Does each activity have a clearly defined deliverable?
    - What output should the activity produce?
4. Cost and Time Estimate – Is each activity defined in a way that allows a meaningful estimate of its calendar time and cost to completion?
    - Often T&E cost is largely driven by the labor cost, and hence the amount of effort needed to conduct it
5. Acceptable Duration Limits – Most activities should be broken down into tasks which are relatively small compared to the size of the full task
    - This varies by project since testing of UASoS can last from days to years.
6. Activity Independence – Are the activities independent of each and practical?
    - Avoid activities that are too complex, or the other extreme, micromanaging
7. Language – Are the activities defined in a way that would be understood by T&E personnel across of the services?
    - This requires language that can be understood even though the services may use different terminologies for similar tasks

Table 60: Work Breakdown Structure for UASoS T&E

ACTIVITY	% of total test effort hours
1. Test Planning	
1.1 Translate SoS capability to requirements/expectations	
1.2 Define mission scenarios	
1.3 Develop a high-level test strategy for each mission scenario	
1.4 Define the critical operational issues that are complete, testable, and traceable to the mission scenarios or SoS requirements/expectations	
1.5 Define the distinct measures of effectiveness (MOE), suitability (MOS) and performance (MOP) that will show whether the SoS has met its expectations and align them to the critical operational issues they assess	
1.6 Assess reports from systems T&E to understand what has already been completed in testing the individual systems within the UASoS	
1.7 Develop detailed test descriptions including the test objective, critical operational issues per mission scenario, metrics per issue, pass/fail criteria, assumptions and constraints	
1.8 Identify and coordinate the physical resources, human resources and infrastructure needed to conduct tests	
2. Test Readiness Review	
2.1 Review preparation	
2.2 Review conduct	
2.3 Review report	
3. Test Execution	
3.1 Set up test environment	
3.2 Conduct test events to address the test objectives for all constitute systems	
3.3 Collect data	
3.4 Determine amendments that need to be made to test plans and re-execute tests accordingly	
4. Test Analysis and Evaluation	
4.1 Retrieve data collected during testing activities	
4.2 If anomalies are detected, analyze for corrective actions e.g.. detect trends in failure to find threats to the system and evidence of design errors	
4.3 Analyze results of tests to assess how the measurements address the critical operational issues identified	
4.4 Document all deviations from expected test results	
4.5 Prepare and deliver test report and management reports containing a summary of the key information gleaned from analysis activities	
Other	

Chapter 5 – Case Study Results

Information presented in this section represents data collected from a past UASoS test effort. This data was provided by the U.S. Army Operational Test Command. Due to confidentiality reasons, the name of the particular project could not be given, nor was there any description of the required capability or mission scenarios; however, sufficient data was provided to help characterize the UASoS, the T&E process, and the size and cost drivers needed as inputs to the cost estimation model.

UASoS Characteristics

The UASoS consisted of systems from two main operational domains, land and air, in addition to various network and communication systems. There were a majority of newer systems (75%) with fewer older systems (25%), with about 75% manned versus 25% unmanned systems. About 44 stakeholders were involved in the T&E process, the distribution shown in Figure 4. Interestingly enough, there were only a few testers (5) involved in the test process. The majority of stakeholders were the contractors and engineers followed by the program managers and test planners.

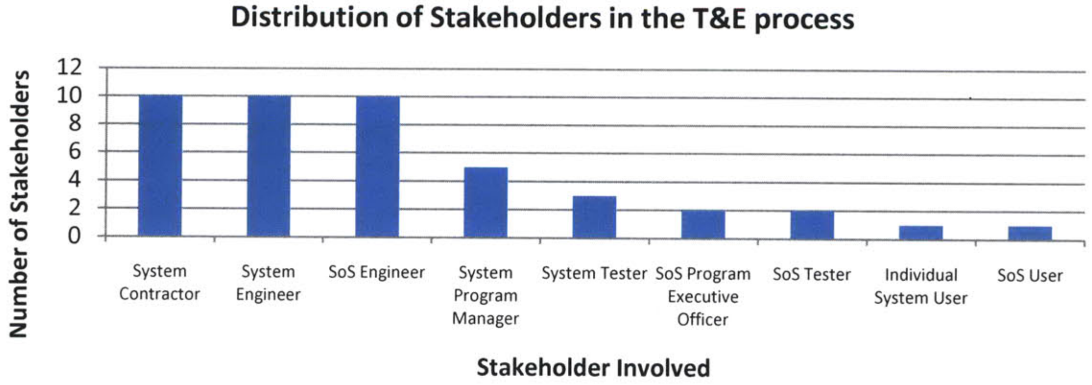


Figure 4: Distribution of Stakeholders involved in UASoS T&E

Test Characteristics

This test effort lasted approximately 6 months, beginning in April 2008, and ending in September that same year, logging a combined 16,000 man hours throughout the process. The entire test was only between 70 and 90% complete, and experienced significant problems, enough to never have such a program repeated in the future. These problems arose during the operational testing phase, which comprised about 90% of all testing done. No system-level operational tests were performed and there was a clear distinction between the developmental and operational testing for this project, the primary reason being that by law, operational testing is conducted by the Operational Test Command whereas developmental test is conducted by any organization the developer chooses.

Approximately 85% of the test was focused on the test planning phase, 3% on the test readiness review, only 2% on the actual test execution, and 10% on the test analysis and evaluation. This data confirm that the majority of the test process is concentrated on the test planning phase which includes tasks ranging from initially identifying and coordinating requirements, to identifying and coordinating resources to execute the test.

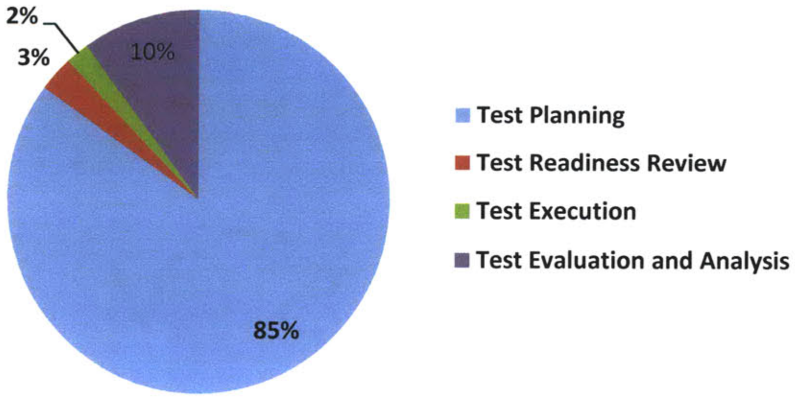


Figure 5: Distribution of test hours throughout the Test and Evaluation of UASoS

The breakdown of the total test effort for each of these tasks within the test planning phase (85% of total effort) is shown in Table 61 below.

Table 61: Percent of total test hours documented for the Test Planning phase

Test Planning Tasks	% of total test effort hours
Translate SoS capability to requirements/expectations	10%
Define mission scenarios	5%
Develop a high-level test strategy for each mission scenario	5%
Define the critical operational issues that are complete, testable, and traceable to the mission scenarios or SoS requirements/expectations	5%
Define the distinct measures of effectiveness (MOE), suitability (MOS) and performance (MOP) that will show whether the SoS has met its expectations and align them to the critical operational issues they assess	20%
Assess reports from systems T&E to understand what has already been completed in testing the individual systems within the UASoS	0%
Develop detailed test descriptions including the test objective, critical operational issues per mission scenario, metrics per issue, pass/fail criteria, assumptions and constraints	20%
Identify and coordinate the physical resources, human resources and infrastructure needed to conduct tests	20%
TOTAL	85%

The data show that more than one fifth of all testing hours were dedicated to defining appropriate and distinct measures of effectiveness, suitability and performance to ensure that the SoS has met its expectation, another fifth was used to provide detailed test descriptions including the test objective, critical operational issues per mission scenario, metrics per issue, pass/fail criteria, assumptions and constraints, and another fifth was used to identify and coordinate the physical resources, human resources and infrastructure for test execution. 10% of the time is used to define requirements and expectations upfront, and 5% is used to define mission scenarios, develop high level test strategies for each mission scenario, and define the critical operation issues that are complete, testable, and traceable to the mission scenarios or SoS requirements/expectations. There was no effort spent on checking previous testing that had been done on the individual systems, partly because most of the systems were new and had never been tested before, and partly because the Operational Test Command was primarily focused on SoS level testing as opposed to system level testing. 85% of the entire effort is spent on the test planning phase, because having well defined, detailed, easily understood test plans, in addition to well coordinated and allocated resources at the appropriate time and place, allows for much more seamless and effective execution. The test planners are charged with the responsibility of ensuring that testers, test engineers, users and program managers are all involved in this process to ensure that all critical factors can be accounted for, and contingency measures are put in place should anomalies arise.

After the test planning phase, about 3% of the test effort is focused on the test readiness review phase. The tasks and effort for each task are shown in Table 62. During this phase four times as much effort was spent on preparing for the reviews of the test plans, as well as conducting the review and making reports of the review.

Table 62: Percent of total test effort hours documented for the Test Readiness Review phase

Test Readiness Review	**% of total test effort hours**
Review preparation	2%
Review conduct	0.5%
Review report	0.5%
TOTAL	3%

The actual test execution only comprised of about 2% of the total test hours, which is a very small fraction of the amount of time needed to plan and set up the tests. The various test execution tasks are shown in Table 63. 1% of the effort was used to set up the tests while half of this was used to conduct tests and collect data. This test effort did not include any amendments to test plans to re-execute tests. Tests were conducted as specified in the test plans throughout the program.

Table 63: Percent of total test effort hours documented for the Test Execution phase

Test Readiness Review	**% of total test effort hours**
Set up test environment	1%
Conduct test events to address the test objectives for all constitute systems	0.5%
Collect data	0.5%
Determine amendments that need to be made to test plans and re-execute tests accordingly	0%
TOTAL	2%

About 10% of the total effort was expended during the evaluation phase of the test process as shown in Table 64. Most of this was spent on analyzing results and detecting trends in failure.

Table 64: Percent of total test effort hours documented for the Test Analysis and Execution phase

Test Planning Tasks	**% of total test effort hours**
Retrieve data collected during testing activities	1%
If anomalies are detected, analyze for corrective actions e.g. detect trends in failure to find threats to the system and evidence of design errors	2%
Analyze results of tests to assess how the measurements address the critical operational issues identified	5%
Document all deviations from expected test results	1%
Prepare and deliver test report and management reports containing a summary of the key information gleaned from analysis activities	1%
TOTAL	10%

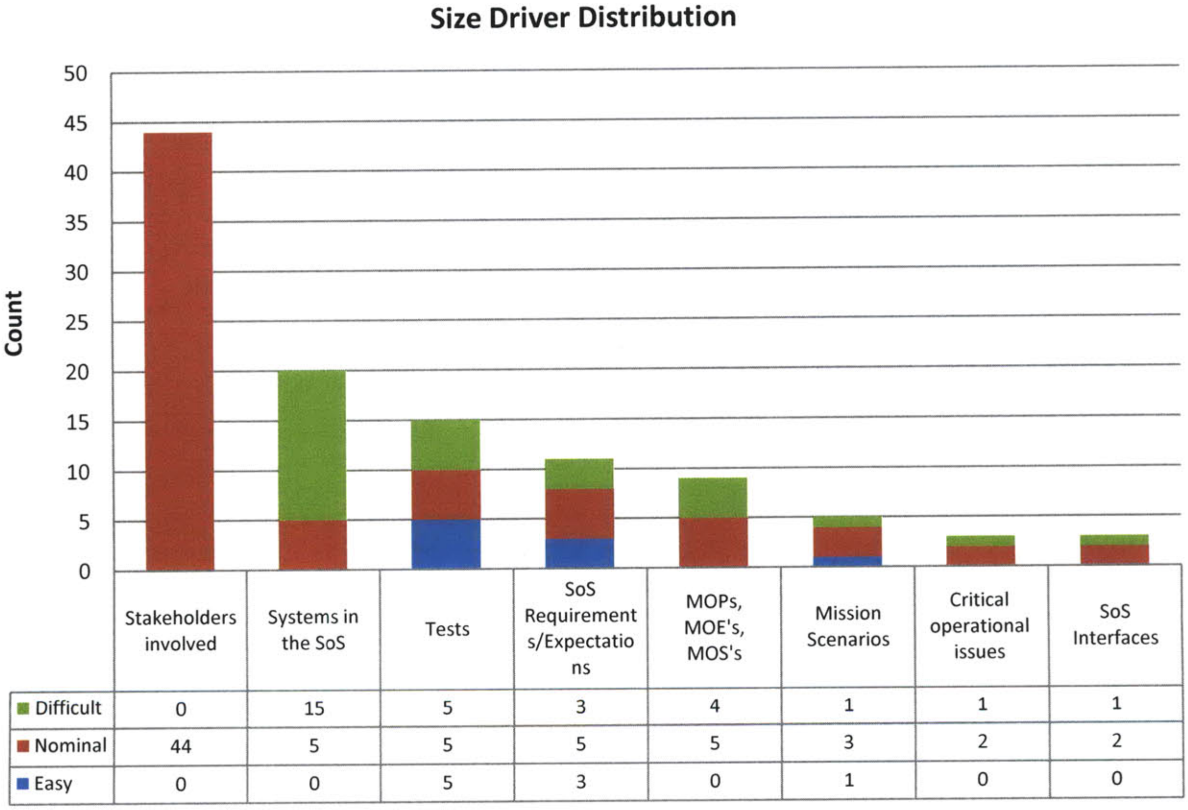
This example is an illustration of the proportions of effort that can exist in a typical test activity. It should be used as a general illustration of what can happen but should not be treated as a universally applicable example since each test activity is unique. For instance, in this data set, there was no effort placed on assessing reports from systems T&E to understand what has already been completed in testing the individual systems within the UASoS. Many operational tests include individual system testing within the SoS environment, and efforts will be made to understand what has already been tested on the system outside the context of the current SoS to determine similarities, differences and possible best practices for approaching new tests. In addition, within the Test Readiness Review phase, there was no effort to make amendments to tests plans, whereas in some test procedures, it is critical to keep testing and readjust test plans accordingly until the UASoS has been sufficiently validated, especially given that emergent behaviors have the potential to make T&E a constant learning process. Translating SoS capability to requirements/expectations, defining mission scenarios, and developing a high-level test strategy for each mission scenario, can also take a larger proportion of the effort, particularly when there are changing requirements, multiple stakeholders and multiple demands.

Size Driver Analysis

The total distribution of size drivers in this UASoS can be seen in Figure 6. Of all the size drivers, the one with the most impact is the number of stakeholders involved. There were 44 stakeholders in all, many of whom were system contractors and engineers, all with some interest in the overall UASoS capability, were aware of each other, and were somewhat aware of each other through the process. There were 20 systems, 15 of which were "difficult": mostly new systems, with new requirements and high levels of autonomy, and low cohesion between them. There were 5 "nominal" ones, which were mostly familiar, with predicable, with moderate cohesion and mostly familiar requirements. An equal distribution of 15 "easy", "nominal" and "difficult" tests, were performed on the UASoS. These represented how well defined the tests were, how easily identifiable they were, the complexity of the requirements, constraints in time and how risky the tests were.

11 SoS level expectations were defined, 5 of which were "nominal". This meant that they were familiar, could be traced to the source as well as to the test objective. 3 of them were very simple to implement, and easily traceable to the objectives. The remaining 3 were very

complex to engineer, were not easily traceable to the source or the test objective and there was a high degree of overlap among them. There were 9 measures of performance, effectiveness and suitability. Of these 4 were "easy": clearly defined and used frequently in the past, easy to identify and resources could be easily found to support the measure. 5 were "nominal", being loosely defined with some degree of overlap. 5 mission scenarios, 3 critical operational issues and 3 main interfaces completed the size drivers in this UASoS.



	Stakeholders involved	Systems in the SoS	Tests	SoS Requirements/Expectations	MOPs, MOE's, MOS's	Mission Scenarios	Critical operational issues	SoS Interfaces
🟩 Difficult	0	15	5	3	4	1	1	1
🟥 Nominal	44	5	5	5	5	3	2	2
🟦 Easy	0	0	5	3	0	1	0	0

Figure 6: Size Driver Distribution UASoS T&E effort

Cost Driver Analysis

For the 20 cost drivers provided, 35 inputs were required since some cost drivers have more than one attribute. Figure 7 gives a distribution of the 35 input ratings provided by the program manager's team. Of these 35 inputs, 40% of the parameters were rated as nominal, 31%

were rated high, 14% were rated very high, 6% were rated very low and very high, and 5% were rated as low impact on cost of UASoS T&E.

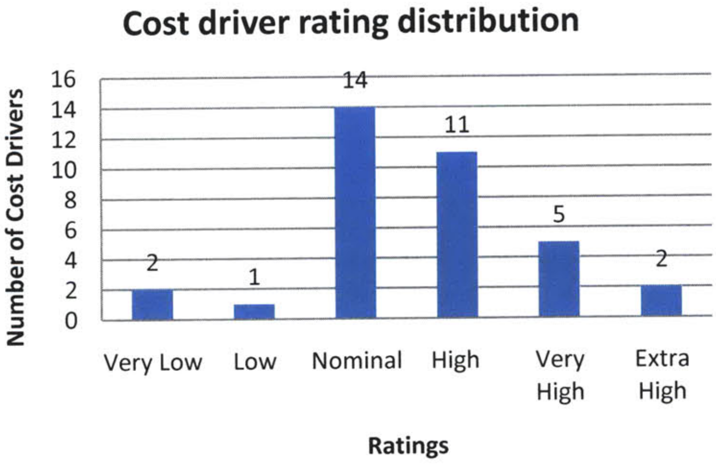


Figure 7: Cost Driver Rating Distribution

The specific ratings of the themes of cost drivers presented in the Cost Driver description section are shown in detail in the remaining sections.

Complexity Factors

Table 65: Complexity Factor Cost Driver Ratings

Cost Driver	Driver Rating
Migration complexity	
Legacy contractor	Nominal
Effect of legacy system on new system	Nominal
System synchronization/integration complexity	
Synchronization: Life Cycle Stage	High
Integration: Technology Maturity	High
Level of Autonomy	High
Technology Risk of SoS Components	
Lack of Maturity	Nominal
Lack of Readiness	Nominal
Obsolescence	Nominal
Test Complexity Level	
Test Maturity	Very High
Test Type	Very High
Test Sensitivity	Very High

Table 65 shows that most of the complexity drivers are either rated as high, very high or nominal. Both attributes of migration complexity were rated as nominal indicating that the legacy systems had been well documented and most of the systems were new. In terms of synchronization complexity, all of the attributes were rated high indicating that the individual systems were at different stages in their life cycles, and effort was spent on making changes to configuration to allow better compatibility. Technology risk was not a major driver of cost here, as the technology had already been proven on pilot projects, the concept had been validated at the appropriate technology readiness level, and no technologies were obsolete. However, the program was faced with unmanned systems that had a high level of human independence, collaborative, high complexity missions and difficult environments in which to perform. In addition, while some tests had been performed in the past, they were very complicated, both hardware and software tests in live and virtual environments were required, and test failure caused system failure with no obvious workaround solution.

Operation Factors

Table 66: Operation Factor Cost Driver Ratings

Cost Driver	Driver Rating
Interoperability of manned and unmanned systems	Very High
Unexpected and undesirable emergent behavior	Nominal
Flexibility	
Technical Adaptability	High
Program Adaptability	High
Synchronization of installations/platforms/tests in the SoS domain	
Sites/Installations	Extra High
Operating Environment	Extra High
Rate of test data collection and analysis	
Frequency	Nominal
Adaptability	Nominal
Documentation match to testing needs	
Formality	Very Low
Detail	Very Low
Level of automation in data collection integration and analysis	Very High

The operation factors ratings were more distributed. Interoperability of manned and unmanned systems was rated as a very high cost driver another indication that the project had a high probability of failure since the UASoS could not successfully demonstrate that it met the requirements or capabilities desired. Minimal unexpected behaviors were seen, however, and the ability to adapt both technically and programmatically as the project progressed was only possible with reconfigurations, which further drove up costs. Synchronization of platforms was rated as an extra high cost driver, because more than 6 installation sites were required and harsh operational environments and security sensitive information was being dealt with. This drove up the costs of daily operations as the project progressed. The level of automation in collecting, integrating and analyzing data also drove up costs because there were mature well integrated tools since the T&E process was built around well state of the art infrastructure. Data were collected as needed so this did not have any particular effect on the cost of the project. In addition, the formality and detail required for reporting was minimal and generalizable enough so that this had a very low impact on cost.

Understanding Factors

Table 67: Understanding Factor Cost Driver Ratings

Cost Driver	Driver Rating
Integration Requirements understanding	Nominal
Architecture Understanding	Nominal

Both integration requirements and architecture understanding were rated as nominal cost drivers, meaning they had no impact on driving costs up or down. They were both reasonably understood, while there was some unfamiliarity with the connections and interoperability of constituent systems, though this was expected going into the test effort and accounted for from the beginning.

People Factors

Table 68: People Factor Cost Driver Ratings

Cost Driver	Driver Rating
Stakeholder team cohesion	
Culture	Nominal
Compatibility	Nominal
Familiarity	High
Personnel/team capability	High
Personnel experience/continuity	
Experience	High
Annual Turnover	Nominal
Test Process capability	High

People factors were rated either as nominal or high impact cost drivers. There was a shared project culture, compatible objectives and a willingness to collaborate; however, the stakeholders had little familiarity with the systems and this drove up costs. Personnel capability was rated at the 75th percentile was meant they had to be trained due to the lack of experience and capability. The stakeholders had been present on past test efforts, but this did not make up for the fact that most of the systems were new and the stakeholders lacked the experience needed for this new project. In addition there was a well defined test process, activities driven by benefit to capabilities, a test focus on the mission scenarios, and a process approach driven by organizational processes which rated test process capability as a high cost driver.

Test Environment Factors

Table 69: Test Environment Cost Driver Ratings

Cost Driver	Driver Rating
Schedule Constraints	
Test Planning	Nominal
Test execution and analysis	Nominal
Testing Resource Challenges	
Availability	High
Allocation	High

Under the test environment factors, neither test planning nor test execution analysis was constrained so they did not have impacts on cost. There were, however, some test resource challenges that rates both availability and allocation as high drivers of cost. Only some resources were available, and there was administrative overhead which created more than necessary costs. The resources were also badly allocated, but replanning was possible, though this still did not help take the project to completion.

Summary

This project was conducted by the U.S. Army Operational Test Command. The appropriate boundaries were drawn around the UASoS only focusing on what was done during the 6 month period at the operational testing level. It comprised more newer systems as opposed to legacy systems with technologies that were not at the highest level of maturity or high enough to make the test process easier. Further, the unmanned system components had a high level of human independence, collaborative, high complexity missions and difficult environments in which to perform. These characteristics could have contributed to the test program not ever being completed, since the issues could not be worked through during the process. While the availability of people and experience were not a major issue, test resource allocation and availability were high drivers of cost. In addition, test complexity was a very high driver of costs. Because this was a new SoS that had never been tested in the past, tests were immature, many different types had to be conducted as the test planners, testers and engineers, were unsure of what the potential risks could be, and the program was very sensitive to test failure. Most attention was placed on the SoS level testing, whereas there needed to have been some effort spend understanding the individual systems in more detail, and figuring out how they interact with each other, as opposed to pushing relatively immature technologies to the operational test environment. More effort could have been spent on simulation environments as opposed to live testing and more effort could have been placed on developing contingency plans upfront, should the test procedures fail. This program was a failure in the test field because the project did not go to completion; however, with better test planning procedures that focused on preparing the individual systems and technologies for SoS level testing, developing contingency plans and prioritizing tests, the program may have been a success.

This information is a sample of the type of data required to continue building and calibrating the model presented in this work. With more data points, a data base of cost estimating relationships can be built from effort distribution, size and cost driver inputs for UASoS T&E.

Chapter 6 – Future Implementation

The previous sections described the methodology used, input parameters, and data required to continue building this preliminary cost estimation model. One of the limitations of parametric cost modeling is the heavy dependence on historical project data and while the use of parametric cost estimation relationships is built on a combination of mathematical modeling and expert judgment, without this data, a reliable model cannot be built. The case study described in the previous section is an indication that data on UASoS T&E do exist, though it will take time and effort, and dedication from stakeholders to provide and collect adequate data to produce a calibrated model.

What is provided in this section is a hypothetical scenario of what can be produced, should there be adequate data to develop and validate reliable cost estimation relationships that meet the criteria outlined in this thesis. Figure 8 outlines the use case for future implementation as part of the new testing infrastructure, PATFrame.

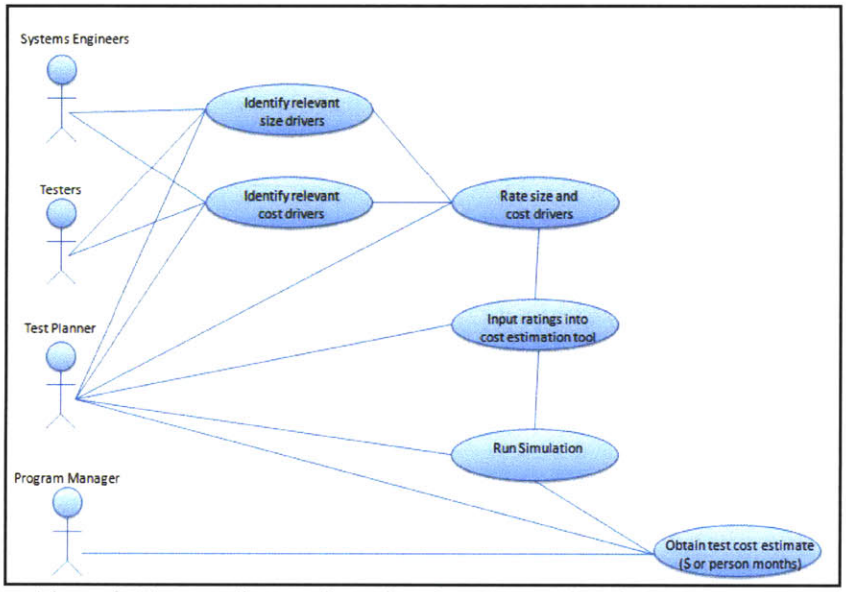
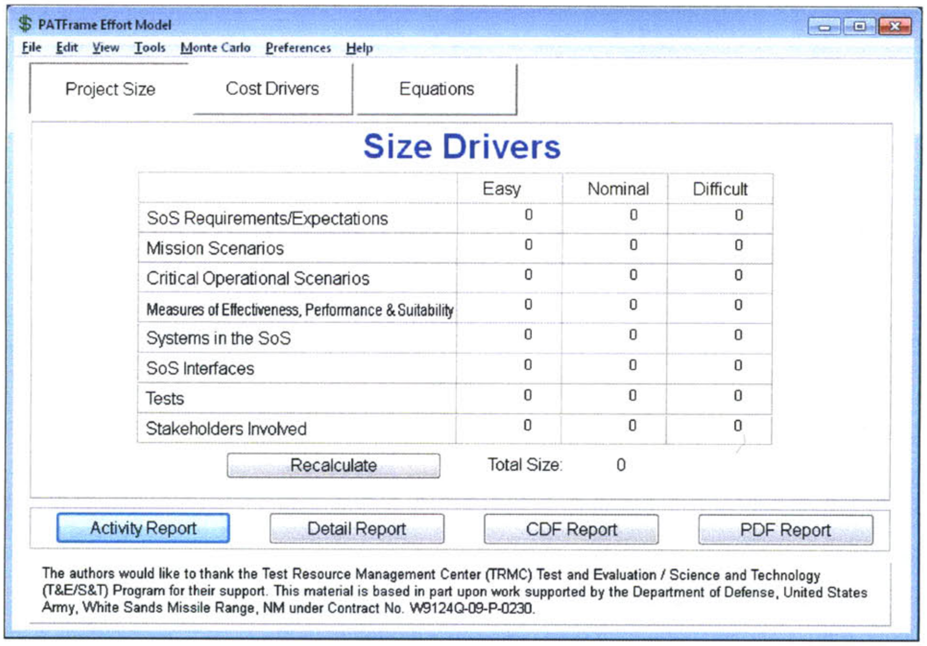


Figure 8: Use case diagram for estimating the cost of UASoS T&E

The goal of this use case (see Appendix C for more detail) is to use parametric cost estimation modeling to determine the effort required for a test in a timely manner to assist with budgeting allocations or reallocations as necessary. A cost approach is used because on a SoS level, there must be a comprehensive analysis of complexity to understand its impact on the cost of systems and to avoid unreliable estimates and unfavorable system performance. This process can also produce strategic options to improve the confidence of cost estimators and stakeholders in making better decisions, even in the face of complexity, risk, and uncertainty.

The main actors using such a software tool are the test planners, testers, program managers, and engineers. Integrated in this tool are cost predicting software and a cost estimation relationship database, which is built through regression models of historical project data. To be qualified to use such a tool, the stakeholders must be knowledgeable of resource availability, time constraints, the characteristics of the participating systems, and knowledge of the size and cost driver ratings specified in the tool. They must also be aware of the execution environment, and understand the capabilities and expectations of the UASoS and the T&E process outcomes. To create a cost estimate, three main steps are involved. First the test planners, testers, and engineers characterize the UASoS, network and test attributes based on provided list of size and cost drivers (Error! Reference source not found. **and** Error! Reference source not found.)**. The graphical user interface UI) they can expect to see is shown below. Using the "help" function (**

Figure 11), each driver is also accompanied by a description containing both its definition and its ratings described in the Model Definition section of this work. The test planner then inputs the ratings for each driver using a simple drop down menu.



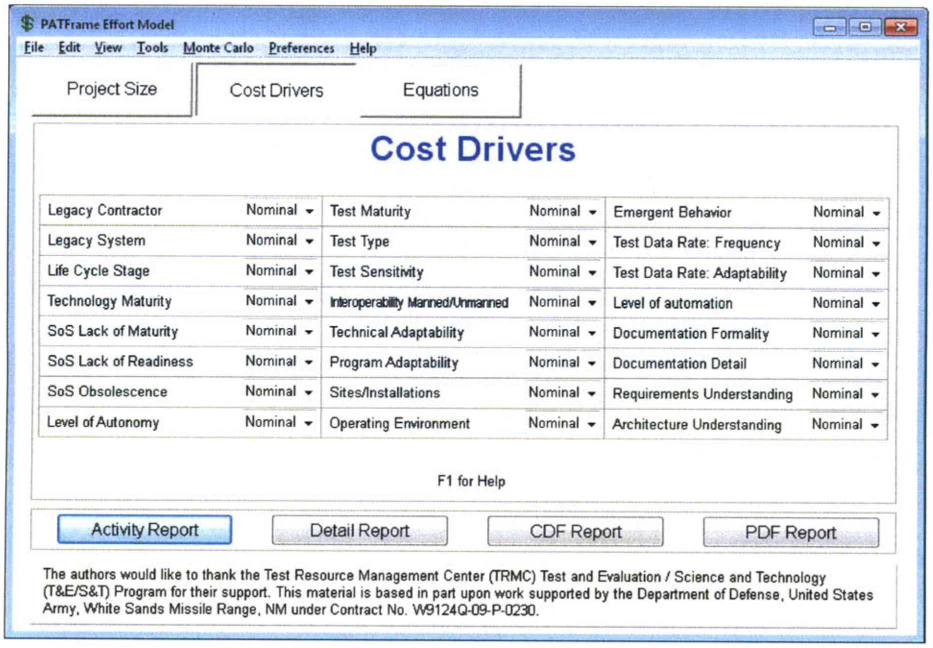
Size Drivers

	Easy	Nominal	Difficult
SoS Requirements/Expectations	0	0	0
Mission Scenarios	0	0	0
Critical Operational Scenarios	0	0	0
Measures of Effectiveness, Performance & Suitability	0	0	0
Systems in the SoS	0	0	0
SoS Interfaces	0	0	0
Tests	0	0	0
Stakeholders Involved	0	0	0

Total Size: 0

The authors would like to thank the Test Resource Management Center (TRMC) Test and Evaluation / Science and Technology (T&E/S&T) Program for their support. This material is based in part upon work supported by the Department of Defense, United States Army, White Sands Missile Range, NM under Contract No. W9124Q-09-P-0230.

Figure 9: Size Driver GUI

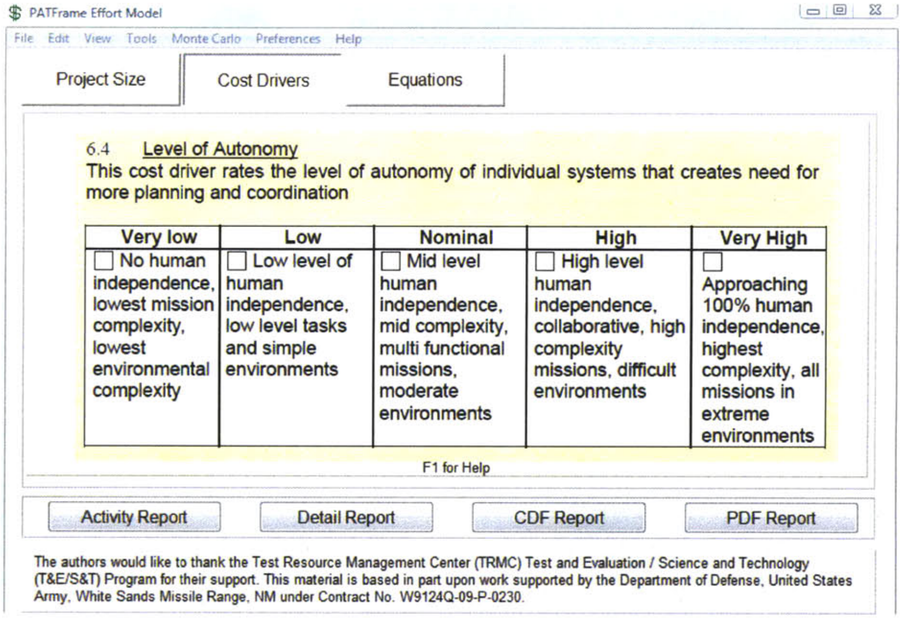


Cost Drivers

Legacy Contractor	Nominal	Test Maturity	Nominal	Emergent Behavior	Nominal
Legacy System	Nominal	Test Type	Nominal	Test Data Rate: Frequency	Nominal
Life Cycle Stage	Nominal	Test Sensitivity	Nominal	Test Data Rate: Adaptability	Nominal
Technology Maturity	Nominal	Interoperability Manned/Unmanned	Nominal	Level of automation	Nominal
SoS Lack of Maturity	Nominal	Technical Adaptability	Nominal	Documentation Formality	Nominal
SoS Lack of Readiness	Nominal	Program Adaptability	Nominal	Documentation Detail	Nominal
SoS Obsolescence	Nominal	Sites/Installations	Nominal	Requirements Understanding	Nominal
Level of Autonomy	Nominal	Operating Environment	Nominal	Architecture Understanding	Nominal

The authors would like to thank the Test Resource Management Center (TRMC) Test and Evaluation / Science and Technology (T&E/S&T) Program for their support. This material is based in part upon work supported by the Department of Defense, United States Army, White Sands Missile Range, NM under Contract No. W9124Q-09-P-0230.

Figure 10: Cost Driver GUI



6.4 Level of Autonomy

This cost driver rates the level of autonomy of individual systems that creates need for more planning and coordination

Very low	Low	Nominal	High	Very High
☐ No human independence, lowest mission complexity, lowest environmental complexity	☐ Low level of human independence, low level tasks and simple environments	☐ Mid level human independence, mid complexity, multi functional missions, moderate environments	☐ High level human independence, collaborative, high complexity missions, difficult environments	☐ Approaching 100% human independence, highest complexity, all missions in extreme environments

The authors would like to thank the Test Resource Management Center (TRMC) Test and Evaluation / Science and Technology (T&E/S&T) Program for their support. This material is based in part upon work supported by the Department of Defense, United States Army, White Sands Missile Range, NM under Contract No. W9124Q-09-P-0230.

Figure 11: Cost driver definition: "Level of Autonomy" in GUI

Finally, the simulation runs and PATFrame produces an estimated effort requirement in dollars and person months based on a number of cost estimating relationships. By inputting the ratings of the size and cost drivers, and using the CER's that relate these drivers, the test planners and program managers can get a cumulative probability distribution of completing the testing and evaluation in a given amount of time shown in Figure 12 below. This tool will calculate the estimated effort required to complete the T&E project for UASoS and the associated probabilities of project completion in that timeframe.

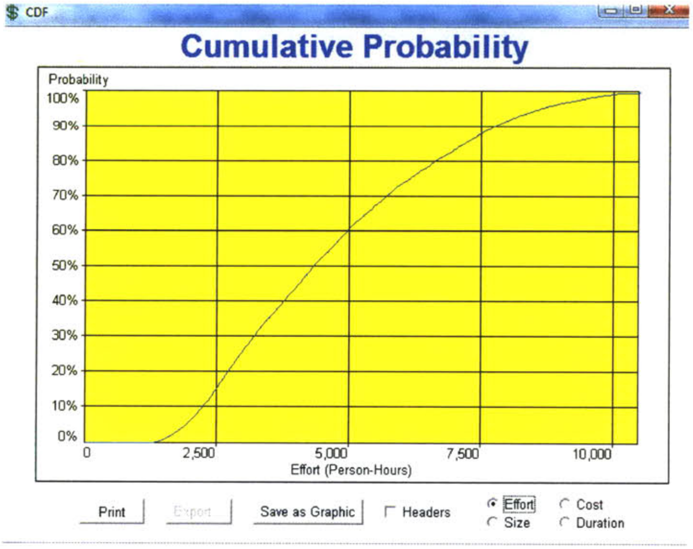


Figure 12: Cumulative Probability Distribution for UASoS T&E Completion

With this information, test planners and program managers should be able to quantify the effort required to test the UASoS, and perform tradeoffs should the need arise or changes occur during the program. These estimates are judged based on the accuracy in effort estimation, as well as the time required to produce these estimates, since there is some work to be done before the drivers are accurately rated. If there are inadequate cost and size driver ratings specified, these will produce errors in the simulation and there will be no results.

However, this tool is far from complete as it stands right now. Because of that quantity of size and cost drivers, adequate data points are required to create regression models to determine the numerical values for each of these drivers and ratings, as well as validate the model form proposed in this work.

Chapter 7 – Policy Implications

Having more reliable cost estimates is only one way to judge whether a test program has been successful or not. A cost estimation model can have significant impacts on how the DoD currently does testing and would help maximize the use of the resources available. For example, think of this hypothetical future scenario. A test planner has a well calibrated mathematical model based on dozens of actual projects and expert judgment at his/her fingertips. He/she inputs the characteristics of the UASoS to be tested by inputting size characteristics and cost drivers ratings with collaboration from the test team. In a matter of seconds they are able to calculate the estimated effort needed to invest in the test program to take it to completion. The model gives them a probability distribution that shows their expected effort. The users tell them they need this UASoS tested in 6 months, but the model says they only have a 50% probability of being done in 6 months. What do they do? They could begin prioritizing their resources and start making alternative arrangements to ensure they can be done in 6 months with a 100% probability. Or if that is impossible, they need to start negotiating with the clients to get more time based on the model projections. They now have some mathematical way of quantifying the risks of UASoS testing, which makes more sense than comparing a current project to a past project that may have insufficient similar attributes to make the comparison worthwhile.

The bottom line: A cost estimation model provides improved analytical capabilities for cost assessment and program evaluation. It is a model based method for calculating effort for test and evaluation and forms a baseline for strategic decision making in DoD acquisition programs. It is also a tool that is built on both expert and historical data, and provides a methodology by which the performance of programs can be monitored. Because current projects are based on similar past projects and extrapolations do not account for the additional risks that could be incurred, using a data base of past projects and correlating that with expert judgment on best practices to create well validated CERS can help bridge that gap. Test planners and program managers are now able to predict the amount of effort they should expect to spend on a program with a greater amount of accuracy.

Further, the size and cost drivers in this model have been built and operationalized with the intent of accounting for the risks of UASoS T&E. There comes a point at which the effort invested in a project will not reduce the risk at a justifiable rate. The intent of this cost model is not to assure the test planner or the user that if a certain amount of effort is invested that all risks have been eliminated. The intent is to predict within a certain probability that a test program can be completed within a certain budget given the assumptions used in characterizing the UASoS and the T&E process. The output effort calculations can be used as guidance on how resources should be allocated, improve a program manager's confidence in estimates, and provide a basis on which prioritizing of resources can occur. It also provides a foundation for strategic decision making to avoid unfavorable system performance. After all, finding problems before delivery, is much cheaper and less time consuming than the alternative.

This cost model will also afford a paradigm shift from allocating resources and then deciding on costs, to prescribing what the possible costs could be and then deciding on how much should be allocated to the program. This, of course, can have both positive and negative effects. If the model overestimates the cost, then the project will have a cost under run; however, if the model is an underestimate, there is the risk of a program being prematurely deemed a budget failure. Macwillie (2010) in his memo said that test costs are a crucial element of operational testing and organizations need to become stewards of public resources and provide other agencies the ability to plan and execute testing with as much transparency as possible. A cost model could offer this transparency by increasing the efficiency in analyzing national security planning and the allocation of defense resources. Multiple programs can be analyzed, planned and coordinated with the click of a button and program managers can coordinate resources with greater ease. Dr. Ashton Carter's memorandum on "Better Buying Power: Guidance for Obtaining Greater Efficiency and Productivity in Defense Spending", emphasized the "Do more without more principle" (Carter, 2010). Program managers now need to treat affordability as a key performance parameter in an effort to conduct a program at a cost constrained by the maximum resources that the department can assign to the capability, which requires programs to use methods to minimize their cost and schedules as effectively as possible. A cost model could be a tool that will help program managers allocate resources where they are most needed, reorganize resources to prevent wastage in any areas, and ensure that the most value is extracted from the effort that is put in.

However, for there to be any success in adopting a new testing infrastructure that includes more reliable cost estimation in DoD UASoS T&E, there needs to be commitment and a cultural shift in mindset from all stakeholders, especially leaders. The characteristics of the model developed and presented need to be understood and predicted as accurately as possible. This requires that everyone play an active role in ensuring testers and engineers are integrated in the test planning process, without relying solely on program managers to do majority of the heavy lifting in predicting the amount of effort needed to conduct tests. This can be done in a number of ways outlined below.

Adopting an Enterprise View to T&E Transformation

Adopting an enterprise view of T&E would be beneficial in analyzing stakeholder values and addressing how the "T&E enterprise" is capable of delivering value to the stakeholders. Such an approach also addresses the envisioned future state of the enterprise, looks at the strategic objectives, aligns current processes with their abilities to achieve those objectives and meet the stakeholder needs, and identifies metrics that can be used to transform the enterprise. The following is a list that is highly recommended for this approach (Murman et al., 2002):

- Adopting a holistic approach to enterprise transformation – Primary stakeholders and value streams must be recognized and understood to ensure value deliver to all stakeholders. These stakeholders include the users, test engineers, test planners, program managers, contractors, individual system owners, leadership, TRMC etc.
- Securing leadership commitment to drive and institutionalize enterprise behaviors – The changes necessary in order to deliver the value to the UASoS T&E community will have to start at the top of the leadership chain, which sets the initial direction of changes that spills off to the other stakeholders in the enterprise.
- Identifying relevant stakeholders and determining their value propositions – This includes conducting an analysis of stakeholders and showing the tradeoffs between their relative importance to the enterprise and their relative. However this will be ongoing as stakeholder needs and values may change with time and program.
- Focusing on enterprise effectiveness before efficiency – More effective procedures to cater for risks such as emergent behaviors need to be developed before shortcuts can be made to get programs fielded in time.
- Addressing internal and external enterprise interdependencies – Placing a boundary on the enterprise is crucial otherwise the problem with be never ending. This includes the

inputs, outputs, internal sequences, internal and external feedback loops, which all need to be well established to deal with all issues accordingly.

- Ensuring stability and flow within and across the enterprise – Creating the ability to identify and remove risk requires clean and easy to follow process flows to clearly assess the inputs, outputs and loops.
- Emphasizing organizational learning –Leadership and personnel need to be aware of the tools available to them, and create an ecology of learning that pushes organizational development through knowledge exchange, documentation, sharing of best practices, and appropriate and undisturbed feedback loops.

Developing a Strategic Foresight for T&E Procedures

Strategic action, as a forward-looking policy that calculates opportunities and threats, is part of the state's core tasks. However, the anticipation, analysis, and interpretation of future developments constitute major challenges (Center For Security Studies, 2009). Strategic foresight is designed as a way of gaining a more comprehensive analysis of what the future may look like and to display the results of such an analysis in a broad array of alternative future scenarios. Such an approach can be used in developing the test procedures that are needed to meet the demands of current UASoS. Initially, the focus would be on gaining information from trends in UASoS risks and emergent behaviors, developments in technology to assist with tests, organizational changes within the DoD etc. These can be used to gauge the early warnings about important developments in new UASoS and could help avoid surprise so that decision-makers have some time for contingency planning. The information will then be processed, interpreted, and the probability of these variations are determined, then various options for action are developed. For example, a particular test procedure can be introduced to deal with a specific risk, or a subject-matter-expert is brought in to evaluate any occurrences of such issues.

Strategic foresight can trigger changes in thinking, improve the coordination of preferences among stakeholders with T&E, and thus help to bring forth new ideas and visions. Strategic foresight would enhance the DoD's strategic decision making capabilities, its capacity to act, to respond, and may thus ease the planning, development, and implementation of political agendas. However, a number of things are required for this to be successful:

- The knowledge accumulated by corporations, think-tanks, academia, and civil society must be utilized and integrated into the foresight process, and not just limited to DoD personnel, as experts both inside and outside the DoD can have great insights into T&E.
- Foresight must be based on reliable and credible sources. If they are not, this leaves recommendations vulnerable to scrutiny and change too quickly.
- There must be sufficient freedom to be creative in thinking of solutions especially since this is a problem that requires "out of the box" type thinking. Strategic foresight is specifically designed to challenge conventional thinking and stimulate innovation.

Expert Analysis to Create Better T&E Infrastructure

Because many of these UASoS are being fielded for the first time and there are not many existing experts on the "T&E procedures for UASoS", this will be an ongoing, dynamic process in which best practices play a key role. The quality of information would evolve over time as expert knowledge improves, programs secure legitimacy and the initial formulation begins to face diminishing returns. The identified tradeoffs are not fixed but will realign slowly as the information-gathering process that is launched by identifying a problem and by taking policy action stimulates increased attention to the problem by experts and diffusion of at least some expert concepts into lay and legal reasoning. A balance needs to be maintained and a ranking of experts would have to be developed to determine how opinions will be weighted.

Heuristics would be very helpful in cases such as these, which depend on actual experimental data to determine emergent properties, etc. Psychologists have found that highly complex decision procedures may not only be ineffective but may well prove counterproductive, even immobilizing. For simple, linear problems, heuristics will work remarkably well; even for more complicated problems, heuristics may be successful over certain periods of time or ranges of explanatory variables.

Experts are another way of having internal feedback loops because they can report what has been going well or not, and help develop best practices and performance indicators. These experts would most likely be stakeholders in the T&E enterprise, ranging from contractors, to test planners, test engineers and users. Who these experts are, is a question that will probably only be answered with experience and time. There is however, a more immediate need for a clearer path forward for experts to share their ideas and opinions especially when it comes to best practices. Currently, the DoD is not known for having a culture that promotes creativity in solving problems but is more strict with its documented procedures. T&E procedures need to adopt the best practices from UASoS testing and the DoD needs to adopt a culture that allows knowledge exchanges, sharing of ideas and information diffusion across the DoD. Perhaps moving its experts around the various services or across projects would help to spread that experience, or focusing on the most relevant procedures for a particular test may be more helpful as opposed to a document filled with extraneous information. Development of a better knowledge base in the form of UASoS test documentation using inputs from stakeholders especially the experts and the users, would also be very beneficial.

Development of Appropriate Metrics and/or Performance Indicators

Because not all pros and cons of an issue can necessarily be measured on the same scale, it is necessary to develop methodologies to assist with predicting what is going well and what isn't. However, as an example for T&E, what is unsafe for one particular UASoS in one environment may not be the same evaluation of risk of the same UASoS in another environment. Performance indicators are important to T&E for two main reasons: their ability to sense and control the T&E process. Some plausible metrics but surely not exhaustive set of metrics are given below along with potential flaws.

- Time required to model a UASoS and its characteristics : This, of course, differs from one program to the next and would be very difficult to create a baseline for programs since they are so different.
- Accuracy of identifying emergent behavior: If emergent behavior are "emergent" how is it possible to determine the level of accuracy?
- Speed with which a program gets through the T&E process: Again, this would be different from one program to the next and the ability compare across programs to create a baseline and knowledge base for T&E globally would be difficult.
- Number of programs that are tested in a period of time: This can create incentives for shortcuts so that bases compete for who has the most output.

However, it is important to note that while adopting performance measures can provide incentives to reduce inefficiencies, the tradeoff is that it discourages excelling in areas that are not measured or incentivized. Particularly when the outcome measures themselves may be in flux, the definition of performance itself is a legitimate subject of debate. In such cases, a broader picture of whether the program has met the capability or not, may be helpful.

Chapter 8 – Conclusion

This thesis began the development of a parametric cost model for the test and evaluation of unmanned and autonomous systems of systems, as part of the Prescriptive and Adaptive Testing Framework, which is currently under development. The model will potentially calculate the estimated effort required to complete the T&E project for UASoS and the associated probabilities of project completion in that timeframe. Given the current challenges of UASoS and the need for testing that is effective and focused on recognizing the risks and failure points of the UASoS, there is need for a new testing infrastructure. Because the capability demands of current and future UASoS outweigh the ability of current test to match these capabilities, the PATFrame decision support system helps fill the gaps of lack of information, identifies best practices, and enhances the ability to adapt to changes as UASoS become more complex.

Macwillie (2010) in his cost estimation memo said, that test costs are a crucial element of operational testing and organizations need to become stewards of public resources and provide other agencies the ability to plan and execute testing with as much transparency as possible. A cost model could offer this transparency by increasing the efficiency in analyzing national security planning and the allocation of defense resources. Multiple programs can be analyzed, planned and coordinated with the click of a button and program managers can coordinate resources with greater ease. Dr. Ashton Carter's memorandum on "Better Buying Power: Guidance for Obtaining Greater Efficiency and Productivity in Defense Spending", emphasized the "Do more without more principle" (Carter, 2010). Because program managers now need to treat affordability as a key performance parameter, this requires programs to use methods to minimize their cost and schedules as effectively as possible. A cost model is a tool that will help program managers allocate resources where they are most needed, reorganize resources to prevent wastage in any areas, and ensure that the most value is extracted from the effort that is put in.

In parametric cost modeling, cost estimating relationships are critical for cost prediction, and are built from regression analysis. The parameters used to build these CER's are developed and operationalized in this thesis using a combination of subject matter expert opinion, literature review and historical project data. Definitions and rating scales are designed for each of the 8 size drivers and 20 umbrella cost drivers. In addition, because this model is designed to include the test procedures across all the DoD services, there is need to have a common language and a baseline from which to characterize the test process. A generalized work breakdown structure was created to help create this baseline and provide a boundary within which test effort could be calculated.

Because the success of parametric cost models is very data driven, this thesis did not take the model to completion. A case study of data collection was presented, and project information was provided by the Army Test and Evaluation Center to characterize the UASoS, the T&E process, and the size and cost drivers needed as inputs to the cost estimation model. The case study is an indication that data on UASoS T&E do exist, though it will take time and effort, and dedication from stakeholders to provide and collect adequate data to produce a calibrated model.

The data also showed that 85% of the total test process is concentrated on test planning. The cost estimation model could potentially be made into a software tool that will be part of the PATFrame decision support system to assist with test planning. The main actors using such a software tool are the test planners, testers, program managers, and engineers. Stakeholders must be knowledgeable of resource availability, time constraints, the characteristics of the participating systems, and knowledge of the size and cost driver ratings specified in the tool.

They must also be aware of the execution environment, and understand the capabilities and expectations of the UASoS and the T&E process outcomes. To create a cost estimate, three main steps are involved. First the test planners, testers, and engineers characterize the UASoS, network and test attributes based on provided list of size and cost drivers. The test planner then inputs the ratings for each driver using a simple drop down menu. Finally, the simulation runs and PATFrame produces an estimated effort requirement in dollars and person months based on a number of cost estimating relationships. By inputting the ratings of the size and cost drivers, and using the CER's that relate these drivers, the test planners and program managers can get a cumulative probability distribution of completing the testing and evaluation in a given amount of time. With this information, test planners and program managers should be able to quantify the effort required to test the UASoS, and perform tradeoffs should the need arise or changes occur during the program.

However, for there to be any success in adopting a new testing infrastructure that includes more reliable cost estimation in DoD UASoS T&E, there needs to be commitment and a cultural shift in mindset from all stakeholders, especially leaders. The characteristics of the model developed and presented need to be understood and predicted as accurately as possible. This can be done through adopting an enterprise view to T&E transformation, developing strategic foresight for T&E procedures, soliciting more expert advice and best practices to create a better T&E infrastructure, and developing appropriate metrics and/or performance indicators.

A cost estimation model can have significant impacts on how the DoD currently does testing and would help maximize the use of the resources available. It is a model based method for calculating effort for test and evaluation and forms a baseline for strategic decision making in DoD acquisition programs. Because current projects are based on similar past projects and extrapolations do not account for the additional risks that could be incurred, using a data base of past projects and correlating that with expert judgment on best practices to create well validated CERS can help bridge that gap. Test planners and program managers are now able to predict the amount of effort they should expect to spend on a program with a greater amount of accuracy. This cost model will also afford a paradigm shift from allocating resources and then deciding on costs, to prescribing what the possible costs could be and then deciding on how much should be allocated to the program. The intent is to predict within a certain probability that a test program can be completed within a certain budget given the assumptions used in characterizing the UASoS and the T&E process. It also provides a foundation for strategic decision making to avoid unfavorable system performance.

References

Aranha, E., & Borba, P. (2007). Sizing System Tests for Estimating Test Execution Effort. *Forum American Bar Association.*

Boehm, B., & Basili, V. R. (2001). Software Defect Reduction Top 10 List. *Software Management*, *34*(1), 135-137.

Boehm, B. W., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., et al. (2000). *Software Cost Estimation with Cocomo II* (p. 544). Prentice Hall.

Boehm, Barry, Valerdi, R., Lane, J. A., & Brown, A. W. (2005). COCOMO Suite Methodology and Evolution. *CrossTalk - The Journal of Defense Software Engineering.*

Carter, A. (2010). *Better Buying Power: Guidance for Obtaining Greater Efficiency and Productivity in Defense Spending. Technology.*

Center For Security Studies. (2009). Strategic Foresight: Anticipation and Capacity to Act. *CSS Analyses in Security Policy*, *52*.

Dahmann, J., Rebovich, G., Lane, J. A., & Lowry, R. (2010). Systems of Systems Test and Evaluation Challenges.

Deonandan, I., Valerdi, R., & Lane, J. A. (2010). Cost and Risk Considerations for Test and Evaluation of Unmanned and Autonomous Systems of Systems. *Work* (pp. 1-6).

Department Of Defense. (2007). UNMANNED SYSTEMS SAFETY GUIDE FOR First Edition. *Defense*, (January).

Dixit, I., & Valerdi, R. (2007). Challenges in the Development of Systems Engineering as a Profession. *International Council on Systems Engineering*. San Diego, California.

DoD. (2008a). *Systems Engineering Guide for Systems of Systems. Technology.*

DoD. (2008b). Department of Defense Intruction 5000.

George, E. (2010). Managing your Way through the Integration and Test Black Hole. *Test.*

Hess, J., Cowart, K. K., Deonandan, I., Kenley, C. R., & Valerdi, R. (2010). Normative and Descriptive Models for Test & Evaluation of Unmanned and Autonomous Systems of Systems. *International Council on Systems Engineering* (pp. 1-11).

Krygiel, A. J. (1999). *Behind the Wizard's Curtain: An Integration Environment for a System of Systems* (p. 274).

Lane, J., & Boehm, B. (2006). Synthesis of Existing Cost Models to Meet System of Systems Needs. *Conference on Systems Engineering Research* (pp. 1-10). Retrieved from http://sunset.usc.edu/csse/TECHRPTS/2006/usccse2006-611/usccse2006-611.pdf.

Lane, J. A. (2009). *Impacts of System of System Management Strategies on System of System Capability Engineering Effort. Engineering.*

Li, Q., Li, M., Yang, Y., Wang, Q., Tan, T., & Boehm, Barry. (2009). Bridge the Gap between Software Test Process and Business Value: A Case Study. *Development Processes* (pp. 212-223). Retrieved from http://www.springerlink.com/index/b43507l0664h1822.pdf.

Macias, F. (2008). The Test and Evaluation of Unmanned and Autonomous Systems. *ITEA Journal, 29,* 388-395. Retrieved from http://www.itea.org/files/2008/2008 Journal Files/December 2008/jite-29-04-388.pdf.

Macwillie, D. (2010). *Test Cost Estimatess. Policy.*

Morgan, M. G. (1993). Risk Analysis and Management. *Scientific American*, (July).

Murman, E., Allen, T., Bozdogan, K., Cutcher-Gershenfeld, J., McManus, H., Nightingale, D., et al. (2002). *Lean Enterprise Value: Insights from MIT's Lean Aerospace Initiative* (p. 360). Palgrave Macmillan.

Singer, P. W. (2009). *Wired for War* (p. 499).

Tenorio, T. (2010). Meeting the Challenges of Unmanned and Autonomous System.

United States Government Accountability Office. (2008). *Unmanned Aircraft Systems Federal Actions Needed to Ensure Safety and Expand Their Potential Uses within the National Airspace System Highlights.*

Valerdi, R. (2005). *The Costructive Systems Engineering Cost Model (COSYSMO).*

Valerdi, R. (2008). *Pioneers of Parametrics.*

Valerdi, R., & Wheaton, M. (2005). ANSI/EIA 632 As a Standardized WBS for COSYSMO. *Proceedings of the AIAA 5th Aviation, Technology, Integration, and Operations Conference* (pp. 1-6). Arlington, Virginia.

WEAPON SYSTEMS ACQUISITION REFORM ACT OF 2009. (2009). *Public Law.*

Young, L. Z., Farr, J. V., & Valerdi, R. (2010). *Conference on Systems Engineering Research* (pp. 1-10).

Appendix A – Survey Rating Initial Cost Drivers

Your responses in this survey should reflect your personal experiences throughout your career and not be dramatically influenced by a single experience. We are interested in your experiences in operational testing of Systems of System (SoS), specifically with unmanned and autonomous systems. This survey is divided into 4 main sections. We first gather some general information on your experiences with the testing process, your opinions on some proposed technical and organizational cost drivers for testing SoS, and then ask for your general input on the testing of unmanned and autonomous SoS. Survey responses will remain anonymous. Participant information is collected for follow-up purposes only.

Section 1: Participant information

Name:____

Organization: ____

Current position: ____________________________ Years in position: ___________

Email address:_______________________________ Phone #: ______________

Experience Information

1. Have you ever been involved in: (Please indicate years of experience for elements that apply)
_____unmanned & autonomous system testing
_____SoS testing
_____SoS testing with unmanned & autonomous components

2. In what capacity have you been involved in testing? Please indicate years of experience for the elements that apply.
_____ system developer
_____budget allocation
_____data collection
_____range controller
_____tester
_____material allocation
_____data analysis
_____range resource manager
_____test planner
_____schedule planning
_____test engineer

What was your specific task(s)?____
____

3. From what perspective are you approaching this survey?
____experience in operational testing ____experience in developmental testing
____experience with both (if you check this one then you will need to fill out two surveys)

2. In what domain(s) have you been involved in testing? Provide number of years of experience for all that apply.
_____space _____air ____land ____sea ____undersea

3. For what service(s) do you have experience with as an employee, contractor, consultant, etc.? Provide number of years of experience for all that apply.
___Army ___Navy ___Air Force ___Joint Program

For questions contact:
Indira Deonandan
Massachusetts Institute of Technology
Cambridge, MA, 02139.

Section 2: The following are a list of **TECHNICAL DRIVERS** that we believe may influence the cost of test & evaluation of SoS. Please rate these on a scale of 1 (low) to 5 (high) to show how you believe these drivers impact the cost of System of Systems testing.

	Impact on Test & Evaluation costs				
	Low		Med		High
	1	2	3	4	5
System of System Level					
Number of requirements of the SoS					
Number of systems to be integrated					
Changes in the requirements of the SoS					
Number of interfaces in the SoS					
Technology maturity of SoS					
System synchronization complexity					
Integration complexity					
Migration complexity – legacy systems impact migration to new capability					
Diversity of platforms within the SoS					
Interoperability of manned and unmanned systems					
Coordination of system platforms – space, air, land, sea, undersea					
Individual System Level					
Coordination requirements to access systems					
Degree of autonomy of individual systems					
Varying levels of maturity of the technology					
Varying technology readiness level of individual systems					
Network Attributes/Characteristics					
Breakdown in communication links such as bandwidth capability					
Type and complexity of operational environment or scenario					
Number of missions					
Power availability for adapting new technologies					
Testing Attributes/Characteristics					
Level of safety					
Reuse of equipment and infrastructure					
Number of tests					
Diversity of tests					
Complexity of tests					
Maturity level of test					
Rate of test data collection and analysis					
Match of material availability and schedule requirements					
Type of operational testing- live, virtual or constructive?					
Availability of testing infrastructure					
Other – please add drivers you believe are missing or have not been captured adequately by those mentioned					

Section 3: The following are a list of **ORGANIZATIONAL DRIVERS** that we believe may influence the cost of test & evaluation of SoS. Please rate these on a scale of 1(low) to 5(high) to show how you believe these drivers impact the cost of System of Systems testing.

	Impact on Test & Evaluation costs				
	Low		Med		High
	1	2	3	4	5
Security level of the project					
Understanding of the project requirements					
Understanding of integration of requirements (OT-function and DT-design)					
Understanding of the architecture of the SoS					
Number of organizations involved in SoS testing					
Availability of resources to assist with SoS integrated test					
Appropriate allocation of resources to assist with SoS integrated test					
Time constraints					
Test process capability					
Reuse of: - existing test strategies and methods - plans Other?					
Stakeholder team cohesion					
Personnel and team capability					
Experience: people who have done similar testing in the past					
Personnel and team continuity					
Support from test planning tools					
Multisite coordination such as geographic location of systems and people					
Other – please add drivers you believe are missing or have not been captured adequately by those mentioned					

Section 4: **Additional Questions**

1. What is your definition of a SoS?

2. What is the most important guidance document that you currently follow in the test planning and execution processes for your system/SoS?

3. How is test cost estimation and resource allocation currently done within you organization?

4. What is the involvement of the test planners in the test execution of the system or SoS?

5. Who is currently involved in the test design phase – that is figuring out what to test and how to test it?

6. What is the involvement of the testers in the test planning phase – that is logistics, resources and schedules?

7. What are the differences between SoS testing and System testing?

8. What information/resources/tools from past testing efforts can be reused to facilitate the planning and testing of future systems? ______________________________

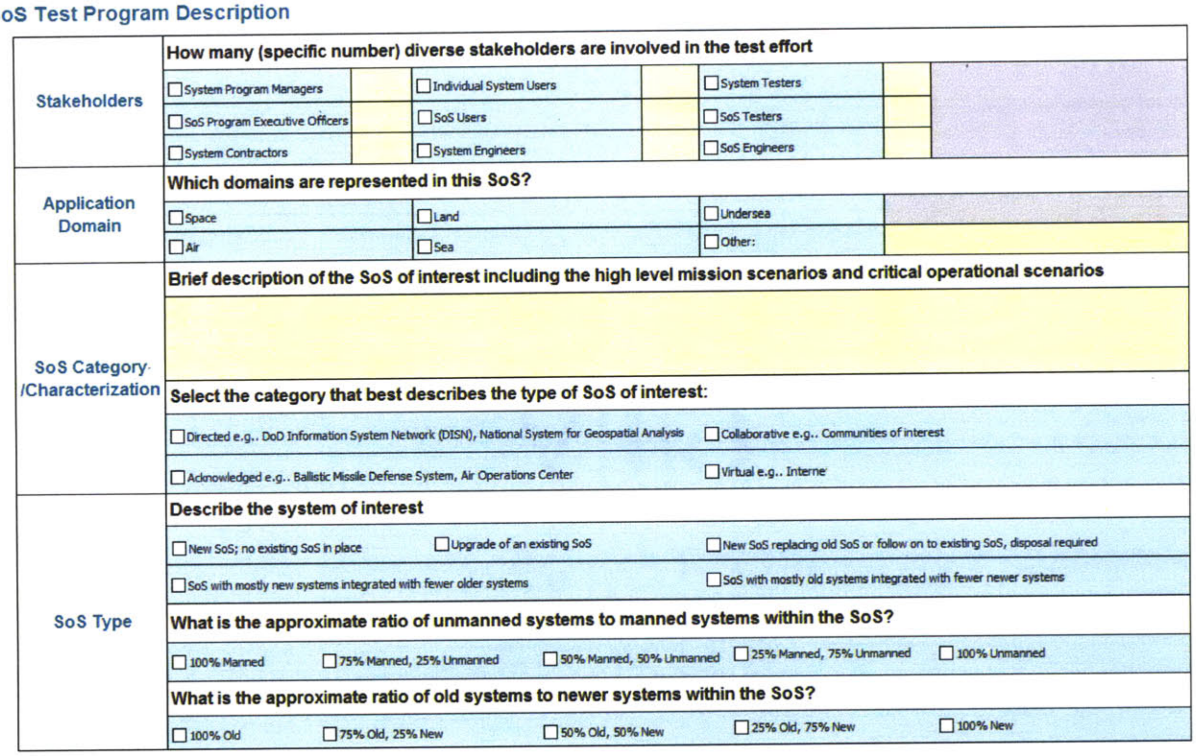
9. Do budget and resource constraints change after testing has begun? _____yes _____no
How do you deal with this? ______________________________

10. Have you explicitly done SoS testing?

11. Please provide any additional comments on the cost drivers presented in the previous two sections.

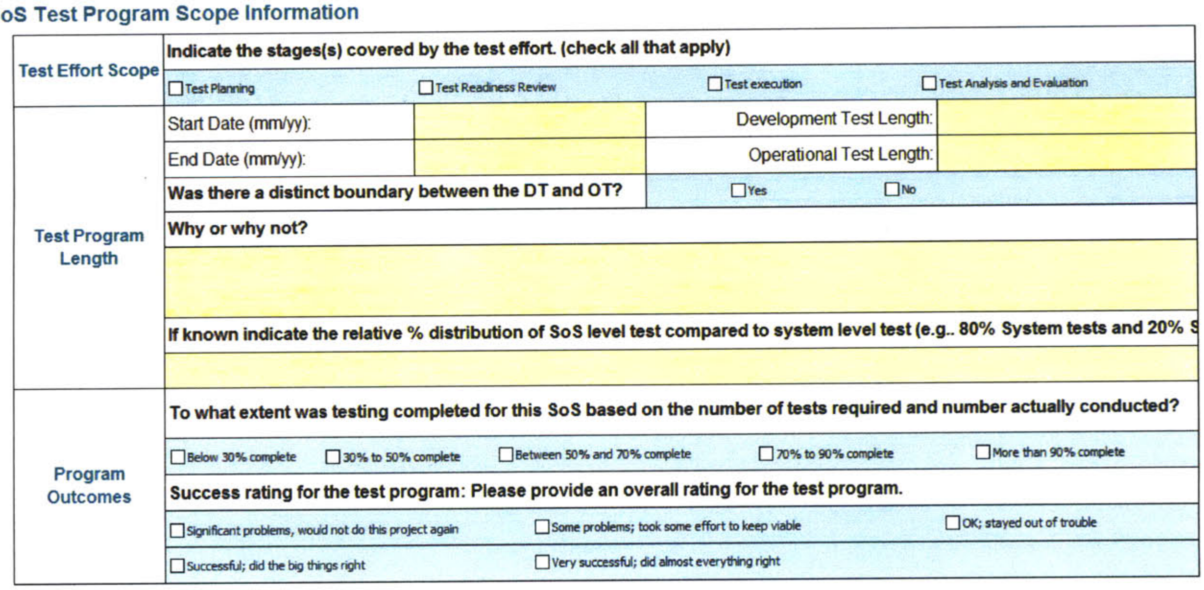
Appendix B – Data Collection Form Snap Shots

UASoS Test Program Description



Stakeholders	**How many (specific number) diverse stakeholders are involved in the test effort** ☐ System Program Managers ☐ SoS Program Executive Officers ☐ System Contractors ☐ Individual System Users ☐ SoS Users ☐ System Engineers ☐ System Testers ☐ SoS Testers ☐ SoS Engineers
Application Domain	**Which domains are represented in this SoS?** ☐ Space ☐ Air ☐ Land ☐ Sea ☐ Undersea ☐ Other:
SoS Category /Characterization	**Brief description of the SoS of interest including the high level mission scenarios and critical operational scenarios** **Select the category that best describes the type of SoS of interest:** ☐ Directed e.g.. DoD Information System Network (DISN), National System for Geospatial Analysis ☐ Collaborative e.g.. Communities of interest ☐ Acknowledged e.g.. Ballistic Missile Defense System, Air Operations Center ☐ Virtual e.g.. Interne
SoS Type	**Describe the system of interest** ☐ New SoS; no existing SoS in place ☐ Upgrade of an existing SoS ☐ New SoS replacing old SoS or follow on to existing SoS, disposal required ☐ SoS with mostly new systems integrated with fewer older systems ☐ SoS with mostly old systems integrated with fewer newer systems **What is the approximate ratio of unmanned systems to manned systems within the SoS?** ☐ 100% Manned ☐ 75% Manned, 25% Unmanned ☐ 50% Manned, 50% Unmanned ☐ 25% Manned, 75% Unmanned ☐ 100% Unmanned **What is the approximate ratio of old systems to newer systems within the SoS?** ☐ 100% Old ☐ 75% Old, 25% New ☐ 50% Old, 50% New ☐ 25% Old, 75% New ☐ 100% New

UASoS Test Program Scope Information



Test Effort Scope	**Indicate the stages(s) covered by the test effort. (check all that apply)** ☐ Test Planning ☐ Test Readiness Review ☐ Test execution ☐ Test Analysis and Evaluation
Test Program Length	Start Date (mm/yy): End Date (mm/yy): Development Test Length: Operational Test Length: **Was there a distinct boundary between the DT and OT?** ☐ Yes ☐ No **Why or why not?** **If known indicate the relative % distribution of SoS level test compared to system level test (e.g.. 80% System tests and 20% S**
Program Outcomes	**To what extent was testing completed for this SoS based on the number of tests required and number actually conducted?** ☐ Below 30% complete ☐ 30% to 50% complete ☐ Between 50% and 70% complete ☐ 70% to 90% complete ☐ More than 90% complete **Success rating for the test program: Please provide an overall rating for the test program.** ☐ Significant problems, would not do this project again ☐ Some problems; took some effort to keep viable ☐ OK; stayed out of trouble ☐ Successful; did the big things right ☐ Very successful; did almost everything right

TEST PROGRAM INFORMATION

How many total testing hours were documented on this project? For this question, you should include both DT and OT test hours.

If available, provide the percent distribution of hours put into the following test effort activities.

Test Planning			Test Readiness Review		
Test execution			Test Analysis and Evaluatio		

How many DT and OT testing hours were documented on this project? If available, provide the percent distribution of hours put into the following test effort activities. This should match your answers the the above question as well.

Test Planning	DT		OT		Test Readiness Review	DT		OT		
Test execution	DT		OT		Test Analysis and Evaluatio	DT		OT		

If available, provide the % distribution of the total hours (as you described above) for the following testing tasks outlined in the work breakdown structure (WBS). These tasks represent the current scope of PATFrame; however, if your project involves a different set of activities please provide additional information at the bottom of the table

ENTER SIZE PARAMETERS FOR SYSTEM OF SYSTEMS

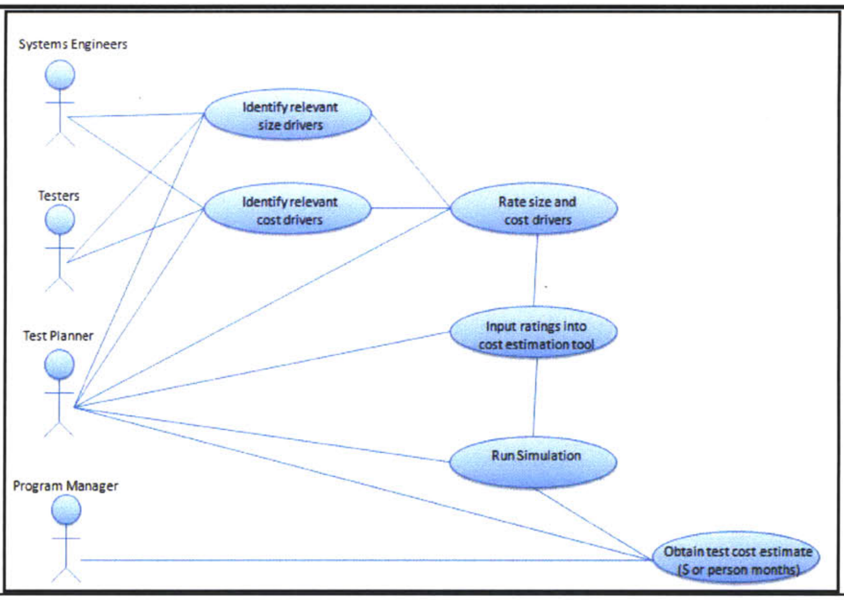
	Total	*Easy*	*Nominal*	*Difficult*	*Uncertainty*	*Source of response*	*Additional Comments*
# of SoS Requirements/Expectations							
How many are reused from previous test activities?							
# of Mission Scenarios							
How many were also identified in previous test activities?							
# of critical operational issues (COI's)							
How many were evaluated in previous test activities?							
# of measures of performance, effectiveness and suitability							
How many have repeated after previous test activities?							
# of systems in the SoS							
How many have been reused from previous test activities?							
How many have been upgraded since the last event?							
# of SoS Interfaces							
How many have been reused from previous test activities?							
How many have been upgraded since the last event?							
# of tests							
How many have been reused from previous test activities?							
How many have been upgraded since the last event?							
# of stakeholders involved							
How many existed in previous test activities?							

SELECT COST PARAMETERS FOR SYSTEM OF INTEREST

	SoS-Relatec	Assumptions/Comments
Migration complexity		
Legacy contractor		
Effect of legacy system on new system		
System synchronization/integration complexity		
Synchronization: Life Cycle Stage		
Integration: Technology Maturity		
Interoperability of manned and unmanned systems		
Level of Autonomy		
Technology Risk of SoS Components		
Lack of Maturity		
Lack of Readiness		
Obsolescence		
Unexpected and undesirable emergent behavior		
Flexibility		
Technical Adaptability		
Program Adaptability		
Synchronization of installations/platforms/tests in the SoS domain		
Sites/Installations		
Operating Environment		
Rate of test data collection and analysis		
Frequency		
Adaptability		
Test Complexity Level		
Test Maturity		
Test Type		
Test Sensitivity		
Schedule Constraints		
Test Planning		
Test execution and analysis		
Testing Resource Challenges		
Availability		
Allocation		
Stakeholder team cohesion		
Culture		
Compatibility		
Familiarity		
Integration Requirements understanding		
Architecture Understanding		
Personnel/team capability		
Personnel experience/continuity		
Experience		
Annual Turnover		

Appendix C– Effort Estimation Model Use Case

Use Case Name:	**Calculate effort for testing unmanned and autonomous SoS**		
Use Case ID:	**5**	Version:	01
Created By:	**Indira Deonandan**	Last Updated By:	Indira Deonandan
Date Created:	5/20/2010	Date Last Updated:	8/04/2010
Update History:			
Goal:	Use a parametric cost estimation model to determine the effort required for a test in a timely manner to assist with budgeting allocations or reallocations.		
Summary:	A cost approach is used because on a SoS level, there must be a comprehensive analysis of complexity to understand its impact on the cost of systems and to avoid unreliable estimates and unfavorable system performance. This process can also produce strategic options to improve the confidence of cost estimators and stakeholders in making better decisions, even in the face of complexity, risk, and uncertainty		
References:	See end of use case description		
Actors:	1. Test planner 2. Tester 3. Program Manager 4. Systems Engineers		
Components:	1. Cost predicting tool 2. Cost estimating relationships 3. Value based testing algorithms		
Trigger:	1. Cost and Size driver ratings 2. Resource availability 3. Time constraints		
Preconditions:	1. Knowledge of resource availability, time constraints, characteristics of participating systems, ratings of size and cost drivers. 2. Execution environment, specified in terms of conditions that can be sensed and the effect of actions that can be taken by the SoS 3. Understanding of the capability and expectation of the SoS		
Postconditions:	1. Quantification of the effort required for testing a particular SoS 2. Ability to perform tradeoffs based on risk and cost		
Normal Flow:	1. Test planner and testers characterize the SoS, network and test attributes based on provided list of size and cost drivers 2. These drivers are rated and ratings are inputs into cost estimation tool. 3. Simulation runs and PATFrame produces an estimated effort requirement in dollars and person months based on a number of cost estimating relationships		
Performance Parameters:	1. Accuracy of effort estimate produced by PATFrame tools 2. Time required to produce effort metric estimates		
Error Conditions:	Not enough size and cost driver ratings specified, or units are inconsistent with what is specified		

Use Case Diagram:	
Notes and Issues:	This is very dependent on the availability of adequate historical data on which to calibrate models

(THIS PAGE INTENTIONALLY LEFT BLANK)