Cost Estimation of Human Systems Integration

by

Kevin K. Liu
B.S., Electrical Engineering
United States Naval Academy, 2008

Submitted to the Engineering Systems Division
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Technology and Policy

at the

Massachusetts Institute of Technology

June 2010

Signature of Author……………………………………………………………………………………………...
Technology and Policy Program, Engineering Systems Division
May 7, 2010

Certified by………..…………………………………………………………………………………………….
Ricardo Valerdi
Research Associate, Lean Advancement Initiative
Thesis Advisor

Certified by………..…………………………………………………………………………………………….
Donna H. Rhodes
Senior Lecturer, Engineering Systems
Principal Research Scientist, Center for Technology, Policy, and Industrial Development
Thesis Advisor

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

Cost Estimation of Human Systems Integration

by

Kevin K. Liu

Submitted to the Engineering Systems Division on May 7, 2010
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Technology and Policy

Abstract

Human Systems Integration (HSI) is the interdisciplinary technical and management processes for integrating human considerations within and across all system elements. The goal of this research is to develop a better understanding of how the costs of doing HSI work within a program can be estimated. The research is divided into two parts.

In the first part, problem formulation, literature from several relevant domains is first reviewed. Next a descriptive case study is conducted on the development of the Pratt and Whitney F119 engine. It examines activities done to support HSI up to engineering and manufacturing development and concludes that, among other factors, HSI in requirements are a major driver of effort. This conclusion leads to work on the integration of HSI into the counting of requirements for an existing systems engineering cost model.

In the second part of the research, implementation and validation, two workshops are conducted to assess how HSI considerations are addressed in real-world requirements engineering. The first workshop tests existing requirements counting guidelines, identifies weakness, and suggests improvement. The second workshop applies the Wideband Delphi method to generate consensus between stakeholders in order to deliver a quantitative estimate of HSI effort. The workshop also demonstrates that stakeholders perceive functional and nonfunctional requirements as driving effort in similar ways, a conclusion that challenges a widely-held belief that nonfunctional requirements are less significant than functional ones.

The research done in the case study and workshops results in improvements to the existing systems engineering cost model, and an application of the model is presented. Policy considerations are discussed. The integration of the HSI into the model represents a significant step toward being better able to plan HSI effort in acquisition programs.

Thesis Supervisors:
Ricardo Valerdi, Research Associate, Lean Advancement Initiative
Donna H. Rhodes, Senior Lecturer, Engineering Systems and Principal Research Scientist,
Center for Technology, Policy, and Industrial Development

Acknowledgments

I owe deep thanks to the many people who have helped me in my two years at MIT.

First, thanks to my advisor, mentor, and friend Dr. Ricardo Valerdi. Thanks to all the people at the Systems Engineering Advancement Research Initiative. In particular, thanks to Dr. Donna Rhodes for your guidance and leadership. Dr. Adam Ross, thank you for tirelessly looking out for the SEArians and caring as much as you do. Thanks to the entire SEAri crew: David Broniatowksi, Deb Chattopadhyay, Kacy Gerst, Dr. Matthew Richards, Nirav Shah, Chris Roberts, Caroline Lamb, Brian Mekdeci, Tsoline Mikaelian, Julia Nickel, Greg O'Neill, Zoe Szajnfarber, and Lauren Viscito. Your insights have been invaluable over the years.

Thanks to the folks at the Technology and Policy Program for creating a truly unique experience. Thanks to Sydney Miller and Ed Ballo for being the pillars who make the program what it is. To all my classmates, thank you for being who you are.

To all the fine Air Force folks at the Air Force Human Systems Integration Office, thanks for making my first "assignment" special and welcoming me into your family. Special thanks for your sponsorship, which allowed this research to happen.

Thanks to my family, who put up with being ignored for the last two years. To Charlene, thank you for agreeing to marry me, that's one less thing to worry about.

Finally, I'd like to acknowledge the United States Marine Corps and United States Naval Academy for giving me the opportunity to spend two years at MIT, an institution I had always dreamed of attending.

Table of Contents

List of Figures

List of Tables

List of Acronyms

AoA	Analysis of Alternatives
CBA	Capabilities-Based Assessment
CCB	Configuration Control Board
CDD	Capabilities Development Document
CER	Cost Estimating Relationship
CICR	Component Integration Change Request
CJCS	Chairman, Joint Chiefs of Staff
COCOMO	Constructive Cost Model
COSYSMO	Constructive Systems Engineering Cost Model
CPD	Capabilities Production Document
DAU	Defense Acquisition University
DCR	DOTMLPF Change Request
DoD	Department of Defense
DOTMLPF	Doctrine, Organization, Training, Materiel, Leadership, Personnel, and Facilities
DTC	Design to Cost
EMD	Engineering and Manufacturing Development
FCB	Functional Capabilities Board
FOC	Full Operational Capability
HCD	Human-Centered Design
HFI	Human Factors Integration
HSI	Human Systems Integration
HSIP	Human Systems Integration Plan
ICD	Initial Capabilities Document
INCOSE	International Council on Systems Engineering
IOC	Initial Operational Capability
IPD	Integrated Product Development
IPPD	Integrated Product and Process Development
IPT	Integrated Product Team
JCIDS	Joint Capabilities Integrated Decision System
JROC	Joint Requirements Oversight Council
KPP	Key Performance Parameter
KSA	Key System Attribute
LCC	Lifecycle Cost
MDA	Milestone Decision Authority
MDD	Materiel Development Decision
MoD	(UK) Ministry of Defense
MSA	Materiel Solution Analysis
O&M	Operations and Maintenance
O&S	Operations and Support
PM	Program Management/Project Management/Program Manager
RDT&E	Research, Development, Test and Evaluation
ROI	Return on Investment
SAF/AQ	Secretary of the Air Force for Acquisition
SE	Systems Engineering
SEMP	Systems Engineering Management Plan
SEP	Systems Engineering Plan
T&E	Test and Evaluation
TD	Technology Development
TOC	Total Ownership Cost
WBS	Work Breakdown Structure

1 INTRODUCTION

1.1 Motivation/Problem Statement

As systems become more complex and software-intensive, the practice of rigorous systems engineering becomes more critical to program performance. The modern practice of systems engineering spans both technical and social disciplines. At the same time, programs are under constant pressure to reduce costs, conform to deadlines, and still deliver required performance.

The study of Human Factors has long been recognized as a critical component of enabling system performance. Existing Human Factors engineering standards and best practices ensure operators, maintainers, and others involved in the lifecycle of a system can interact effectively with the system. However, as systems have become more complex, so have their relationships with the people that interact with them.

Human Systems Integration (HSI) seeks to address the complexities of human considerations in modern systems by integrating multiple human domains into systems engineering early in the development lifecycle. HSI is defined as the "interdisciplinary technical and management processes for integrating human considerations within and across all system elements; an essential enabler to systems engineering practice" (International Council on Systems Engineering 2006).

Despite the prominence given to HSI in a number of policy documents, the National Academies, in a 2007 report on HSI, identified "a lack of commitment by funders and program managers to assign priority to [HSI]" as well as "a lack of effective communication between system engineers and human-system domain experts" to be challenges inhibiting the practice of HSI (Pew and Mavor 2007). As part of its conclusions, the report recommended further research in "estimating the size of the HSI development effort" as a means of achieving "full integration of human systems and systems engineering" (Pew and Mavor 2007).

1.2 Methodology and Methods

1.2.1 Research Questions and Hypothesis

Three research questions were developed from the motivation discussed above, and the research sponsor's requirements. They are:

- R1: How can the "right" amount of effort to invest in HSI be determined?
- R2: How much does HSI effort cost?
- R3: What is the relationship between HSI and systems engineering?

The question of "how much does HSI effort cost?" results from the recommendations of the National Academies report discussed in the previous section. Understanding how much HSI costs within a program can help program managers to plan for HSI and avoid cutting funding for HSI when other priorities arise. However, developing a single value or heuristic about the cost of HSI does not tell the full story of HSI investment. Programs are under constant pressure to reduce costs. HSI investment needs to be "right-sized," that is, enough HSI must be funded to ensure human considerations are integrated into the system, but not so much that additional funding achieves reduced marginal benefit. Therefore, research question R1 asks "how can the 'right' amount of effort to invest in HSI be determined?" The research question "What is the relationship between HSI and systems engineering" reflects the National Academies report recommendation to explore communication between systems engineering and human-system domain experts and to more fully integrate HSI with systems engineering. Understanding how the relationship between the two disciplines will inform how cost estimates developed for HSI can best be integrated with cost estimates developed for systems engineering.

An initial hypothesis was developed to test the research questions:

- H1: Human Systems Integration effort can be estimated as a function of total Systems Engineering Effort.

Subsequent to literature reviews and a case study, a sub-hypothesis was developed to focus implementation and validation work:

- H2: HSI effort can be estimated by counting "number of HSI-related requirements."

1.2.2 Method Selection and Thesis Structure

This thesis draws its research approach from two well-established fields: Human Factors and systems engineering.

Human Factors is an applied science, meaning that it "relies on measurement of behavioral and physical variables" – in particular, the observation of human subjects (Proctor and Van Zandt 2008). Observation in the context of the scientific is known as empiricism. Empirical research provides observations that are used to evaluate "the truth value of alternative statements" (Proctor and Van Zandt 2008).

The need for empirical research in systems engineering has been explored by (Valerdi and Davidz 2009), who recommend using mixed methods as a means to gather enough data to support assertions. Related work has shown that a majority of systems engineering researchers use mixed methods (Valerdi, Liu et al. 2010). The methods used as part of mixed-methods research may be part of an established discipline or may originate from varied disciplines. In the domain of Human Factors research, method selection depends foremost on what observations of human subjects can be made. Runkel and McGrath (1972) recommend different research methods based on, among other factors, access to subjects. The framework they establish (illustrated in Figure 1) was applied in the choosing of research methods in this thesis.

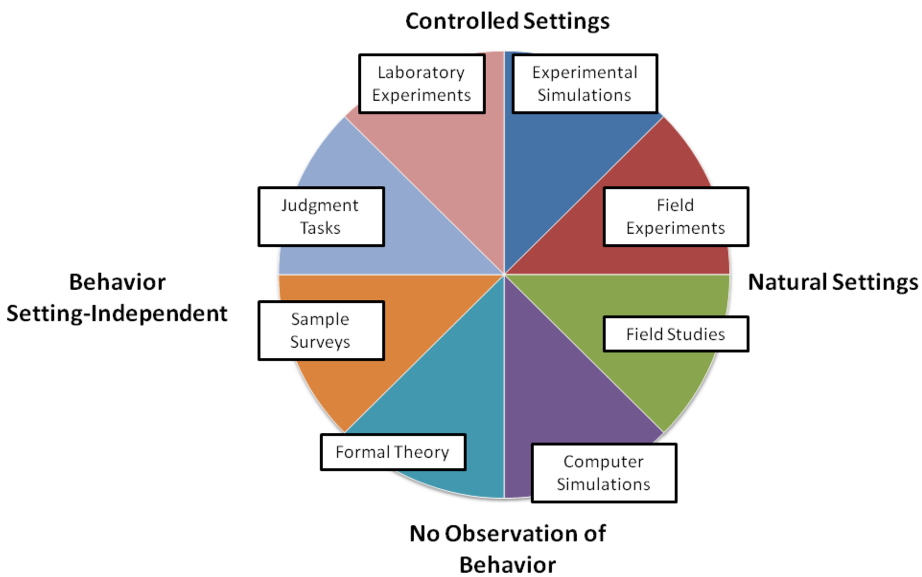


Figure 1. Framework for choosing research methods, adapted from (Runkel and McGrath 1972).

The work in this thesis was divided into two distinct parts: (1) problem formulation, and (2) implementation and validation. Work done during problem formulation was exploratory in nature. A review of relevant literature is discussed in Chapter 2. Chapter 3 describes a descriptive case study (the equivalent of a field study as described by Runkel and McGrath). The case study explored the three research questions in order to refine the hypothesis and guide follow-on work. Insights from the case study resulted in a sub-hypothesis.

The sub-hypothesis developed during problem formulation was tested through two workshops. The workshops are described in Chapter 4, which summarizes implementation and validation work. The experimental simulation method was applied during both workshops in order to maximize the amount of empirical data that could be collected, while taking into account the relatively small number of participants available and the lack of experimental control over participants. Section 4.3 applies the conclusions made during implementation and validation to an example cost modeling application.

Chapter 5 summarizes the research performed during both problem formulation and implementation and validation and shows how the conclusions made contribute to understanding of the research questions and hypotheses.

1.2.3 Related Work

This thesis draws from published work by the author. A list of relevant references can be found in section 6.2. Portions of the text dealing with cost estimation and the COSYSMO model (sections 2.3 and 3.3.1, in particular) draw from (Valerdi and Liu 2010), which in turn uses text from (Valerdi 2008), with the original author's permission.

2 REVIEW OF RELEVANT TOPICS

This chapter presents a review of three topics relevant to the study of HSI: defense acquisition, systems engineering, and cost estimation. The practice of HSI aims to integrate human considerations into systems. Defense acquisition and systems engineering work in concert to realize needed systems. The scope of the review is limited to a broad overview of the U.S. Defense Acquisition System and the role of systems engineering within it, with a focus on how each of these topics relates to HSI. The discussion of cost estimation informs the discussion of existing approaches to HSI cost estimation, discussed in Chapter 3.

2.1 Defense Acquisition

Three key Decision Support Systems work together to support Defense Acquisition. The sum of these processes is commonly referred to as "Big 'A' Acquisition" in order to distinguish the overarching system from the Decision Support System known as "The Defense Acquisition System", commonly known as "little 'a' acquisition" (Department of Defense 2010). Figure 2 shows visually how these terms are related.

The Defense Acquisition System and Joint Capabilities Integration and Development System (JCIDS) together encompass the majority of the policies, principles, and requirements for the development of military systems. The Planning, Programming, Budgeting, and Execution Process deals with how resources are requested, allocated, and tracked within the Department of Defense.

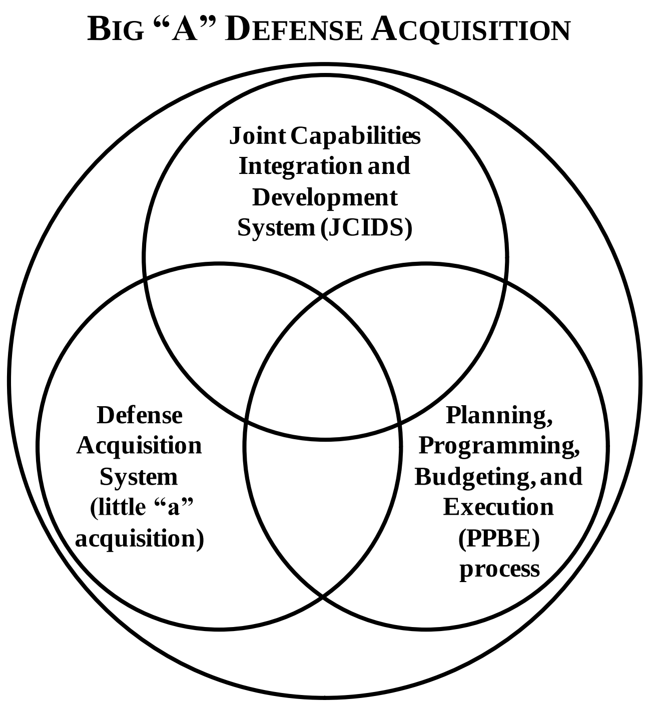


Figure 2. Big "A" Acquisition, adapted from (Defense Acquisition University 2010b).

While defense acquisition policy is written to promote innovation and autonomy, programs must be held accountable to requirements, budget, and schedule. Defense acquisition policy assigns a single individual to each program who is held accountable for that program. The Program Manager (PM) of a program "is the designated individual with responsibility for and authority to accomplish program objectives for development, production, and sustainment to meet the user's operational needs. The PM shall be accountable for credible cost, schedule, and performance reporting" (Department of Defense 2003). The program manager must be familiar with acquisition policy and works to ensure that his/her program meets the requirements of each of the DoD's three Decision Support Systems. The two decision support systems most relevant to HSI are The Defense Acquisition System and JCIDS. These two systems are further discussed in the following sections.

2.1.1 The Defense Acquisition System

The U.S. Defense Acquisition System is defined as "the management process by which the [U.S.] Department of Defense provides effective, affordable, and timely systems to users" and "exists to manage the nation's investments in technologies, programs, and product support necessary to achieve the National Security Strategy and support the United States Armed Forces" (Department of Defense 2003).

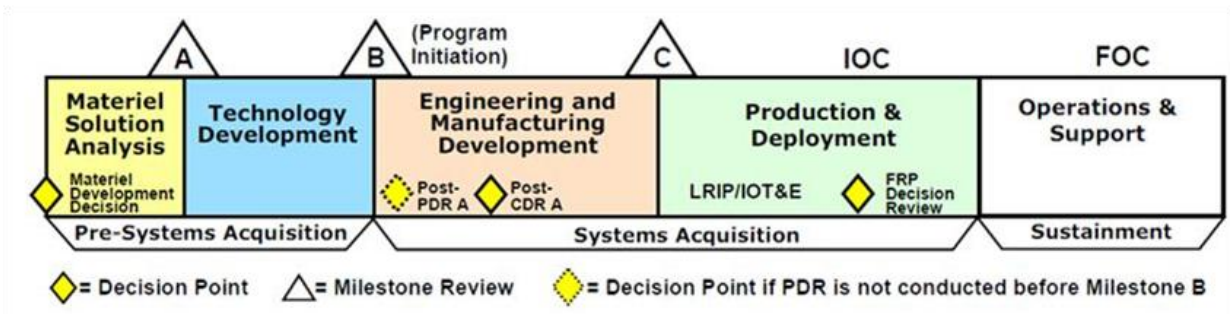


Figure 3. Operation of the Defense Acquisition System (Department of Defense 2008a).

The operation of The Defense Acquisition System (illustrated in Figure 3) is defined by three major milestones and five acquisition phases. The first phase, Materiel Solution Analysis (MSA) exists to "assess potential materiel solutions" (Department of Defense 2008a). When the need for a materiel solution is identified prior to MSA, an Analysis of Alternatives (AoA) is performed during MSA to weigh the pros and cons of possible solutions. The MSA phase is often referred to as "Pre-Milestone A" acquisition because it takes place prior to the major review at Milestone A.

Once a program passes Milestone A Review, it enters the Technology Development (TD) phase of acquisition. The purpose of the TD phase is to "reduce technology risk, [and] determine and mature the appropriate set of technologies" (Department of Defense 2008a). Early prototypes of the system may be developed during this phase, but in general the design and specifications of the system will remain fluid as trades are made between attributes desired and resources available. Taken together, the MSA and TD phases are known as "Pre-Systems Acquisition" or "Pre- Milestone B" acquisition.

After a program passes Milestone B Review, it enters the Engineering and Manufacturing Development (EMD) phase of acquisition. Milestone B also marks "program initiation" because all programs are required to have a program manager after Milestone B, but not necessarily before. By the EMD phase, the requirements of the system should be finalized. The main goal of the EMD phase is to "develop an affordable and executable manufacturing process" taking into consideration systems integration issues including HSI, logistics, supportability, and interoperability. Several types of testing and evaluation (T&E) also occur during EMD.

Although both EMD and its succeeding phase, Production and Deployment (PD), are considered part of "Systems Acquisition" only prototypes can be produced during EMD. In addition, the majority of funds for MSA, TD, and EMD all come from Research, Development, Test, and Evaluation (RDT&E) funding, whereas funds for PD come from Procurement money. The Production and Deployment phase uses the plans developed during previous phases to produce operational units, test initial production units, and then begin full-rate production once tests are passed.

The final phase of acquisition is Operations and Support (O&S). Both O&S and PD are referred to together as "Post-Milestone C" phases. The line between the two program phases is more blurred than the phases that are separated by Milestone Reviews. The goal of PD is to provide Initial Operational Capability (IOC) while the goal of O&S is to provide Full Operational Capability (FOC). However, O&S for the first units produced must often begin before processes that occur during Production and Deployment are complete.

2.1.2 The Joint Capabilities Integration and Development System

Many of the requirements that allow a program to move from one acquisition phase to the next depend on the policies of JCIDS. This is because the documents that govern the Defense Acquisition System (see section 2.1.1, above), known collectively as the DoD 5000 series of publications, were developed to complement Chairman of the Joint Chiefs of Staff Instruction 3170.01G – Joint Capabilities Integration and Development System (Department of Defense 2003) (Department of Defense 2008a; CJCS 2009).

The JCIDS process begins with the establishment of user needs, expressed as capabilities desired. A capability is defined as "the ability to achieve a desired effect under specified standards and conditions through combinations of means and ways across doctrine, organization, training, materiel, leadership and education, personnel, and facilities (DOTMLPF) to perform a set of tasks to execute a specified course of action" (CJCS 2009).

Capability needs, gaps, and excesses are identified through a Capabilities Based Assessment (CBA), which "may be initiated by any number of organizations, to include combatant commands, Functional Capabilities Boards (FCBs), [any of the Armed] Services, and Defense Agencies" (CJCS 2009). CBAs must be linked to strategic security guidance documents such as the National Security Strategy, National Defense Strategy, and National Military Strategy, among others (CJCS 2004; White House 2006) (Department of Defense 2008b). CBAs take place prior to the MSA phase of acquisition and are considered a part of pre-systems acquisition or pre-Milestone A acquisition. The graphic in Figure 4 shows the relationship between key JCIDS documents and phases and milestones of The Defense Acquisition System.

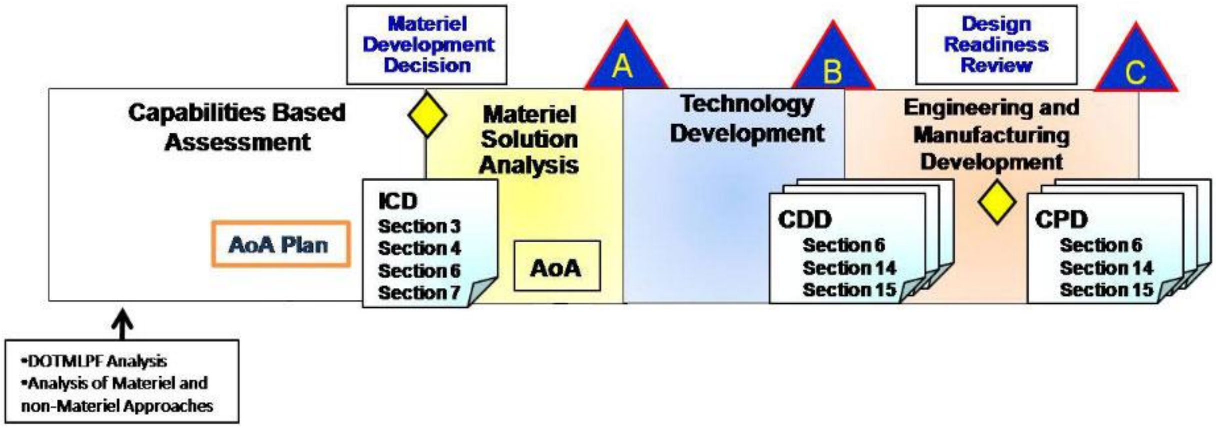


Figure 4. Relationship between JCIDS and the Defense Acquisition System, adapted from (Air Force Human Systems Integration Office 2009b).

CBAs essentially evaluate the need for a materiel solution. CBAs must consider changes to doctrine, organization, training, materiel, leadership and education, policy, and facilities (DOTMLPF) as means to satisfy capability gaps. When a need for a new program emerges, the broad capabilities required of the new program are written into an Initial Capabilities Document (ICD), which is then used to make the Materiel Development Decision (MDD).

Once the MDD has been made, the ICD is used during MSA and TD to refine capabilities required of the system. The next major document produced after the ICD is the Capability Development Document (CDD). CDDs are the "primary means of defining authoritative, measurable, and testable capabilities needed by the warfighters" (CJCS 2009). Much of the work that is performed during MSA and TD center around producing a CDD, and the CDD itself guides the production of other key acquisition documents. A draft CDD is required by Milestone A and a final CDD is required by Milestone B.

The final key JCIDS document, the Capability Production Document (CPD) is produced between Milestone B and Milestone C, during the EMD Phase. CPDs emulate CDDs in both structure and content, the key difference being that whereas CDDs guide the EMD phase of acquisition, CPDs guide the Production & Deployment phase.

2.2 Systems Engineering

The previous sections describe the policies that govern defense acquisition. Systems engineers are the engineers who work within Acquisition to realize systems. This section gives a brief overview of systems engineering, explains the role of systems engineering within defense acquisition, and explores the relationship between HSI and systems engineering.

2.2.1 Definitions and Practice

The practice of systems engineering can be traced back to the mid-1900s post-WWII era, when systems became so complex that many projects began to fail along the lines of performance, budget, and schedule (Ferris 2007). The discipline developed to address these issues by considering systems holistically, viewing them as more than the sum of parts. The first book to begin to define systems engineering was Goode and Machol's '*System Engineering: An Introduction to the Design of Large-Scale Systems* (1957). Since then, many systems engineering standards and definitions have emerged. Some of the most significant are summarized below.

1969: MIL-STD 499: "System engineering is the application of scientific and engineering efforts to (a) transform an operational need into a description of system performance parameters and a system configuration through the use of an iterative process of definition, synthesis, analysis, design, test and evaluation…" (Department of Defense 1969)

1974: MIL-STD-499A: System Engineering is "A logical sequence of activities and decisions transforming an operational need into a description of system performance parameters and a preferred system configuration, (Department of Defense 1974).

1994: MIL-STD-499B: Systems Engineering is "an interdisciplinary approach encompassing the entire technical effort to evolve and verify an integrated and life-cycle balanced set of system people, product, and process solutions that satisfy customer needs"(Department of Defense 1994).

2005: IEEE 1220: "systems engineering is responsible for the total development effort necessary to establish a product design that can be tested, manufactured, supported, operated, distributed, and disposed of" (International Organization for Standardization 2005).

While these standards have helped to shape the practice of systems engineering, there is also a larger question of whether the modern practice of systems engineering transcends traditional definitions. Systems engineering has traditionally employed a set of systems engineering technical processes and technical management processes. Systems engineering standards ISO 15288, *Systems and Software Engineering-System Life Cycle Processes* and ANSI/EIA 632 *Processes for Engineering a System* define these processes (ANSI/EIA 1999; ISO/IEC 2002).

As systems have become more complex, the discipline of systems engineering has necessarily begun to incorporate other relevant fields of study. Rhodes and Hastings argue, "The strongest heritage of Systems Engineering comes from the aerospace and defense industries, and the terminology and language of these industries tends to put artificial boundaries and constraints around it as a discipline and practice" (2004).

The most recent definition of systems engineering put out by the International Council on Systems Engineering (INCOSE) helps to bridge the gap between traditional and advanced approaches to systems engineering. It defines systems engineering as an interdisciplinary approach and means to enable the realization of successful systems," but also emphasizes systems engineering's heritage, emphasizing – "defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem" (International Council on Systems Engineering 2006).

2.2.2 Systems Engineering in Defense Acquisition

As discussed above, systems engineering has its strongest heritage in the defense domain; the Department of Defense requires rigorous systems engineering practice at all levels within its systems. Acquisition programs are required to be "managed through the application of a systems engineering approach that optimizes total system performance and minimizes total ownership costs" (Department of Defense 2003). More specifically, "Systems engineering provides the integrating technical processes to define and balance system performance, cost, schedule, and risk" (Department of Defense 2008a).

Systems engineering plays a role in both The Defense Acquisition System and in JCIDS (both discussed in section 2.1). Every acquisition program is required to have a Systems Engineering Plan (SEP) by Milestone A. The SEP documents "overall technical approach, including key technical risks, processes, resources, metrics, and applicable performance incentives" (Department of Defense 2008a).

Systems engineers facilitate the JCIDS process. They support the generation of the ICD, CDD, and CPD. They develop system requirements and specifications from the CDD and CPD. They ensure that stakeholder requirements are satisfied at multiple points along the lifecycle. Figure 5 sums up the key technical processes systems engineers execute in acquisition programs. The takeaway from the figure is that systems engineering must be supported at every point within a system's lifecycle in order for a system to be realized.

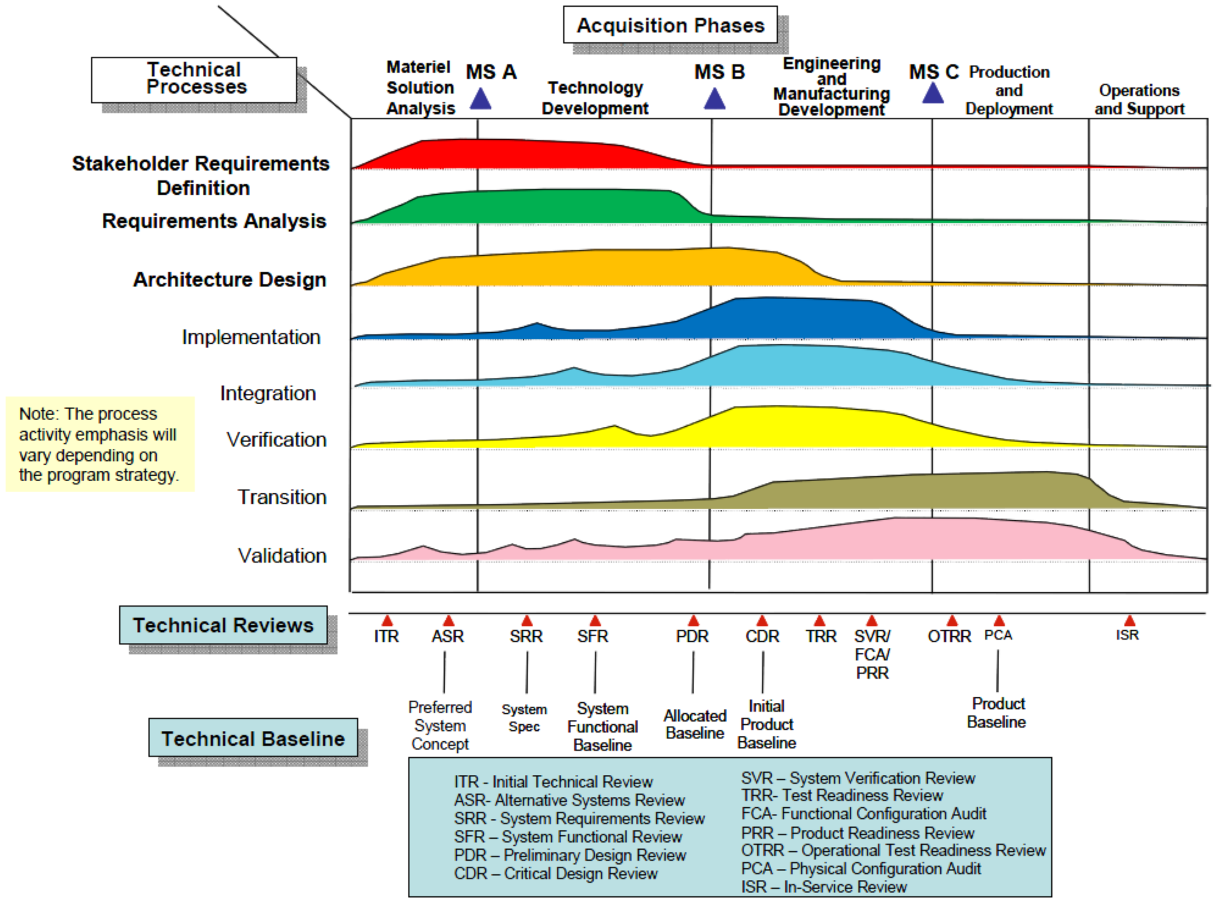


Figure 5. Notional representation of systems engineering technical processes within the Defense Acquisition Lifecycle (Defense Acquisition University 2010a).

2.3 Cost Estimation/Prediction Methodologies and Methods

Cost estimation helps program managers and systems engineers to plan their work, predict costs, and better understand the scope of the systems they develop. Cost estimation is especially important when developing systems of high complexity, cost and duration. The best guidance on cost estimation techniques comes from organizations that have expertise in developing and acquiring these classes of systems. Industry and government guidebooks provide a rich source for best practices, lessons learned, tools and cost estimation processes (Department of Defense 1992; US Army Cost and Economomic Analysis Center 2002; International Society of Parametric Analysts and Society of Cost Estimating and Analysis 2004; Air Force Cost Analysis Agency 2008; National Aeronautics and Space Administration 2008; Government Accountabiility Office 2009).

2.3.1 Analogy

The estimation by analogy method capitalizes on the institutional memory of an organization to develop its estimates. This type of estimate is typically used when only one or very few historical systems similar to the new system exist. The method works best when many similarities between old and new systems exist, as in when a new system is developed using components of previous systems.

Case studies are an instrument of estimation by analogy; they 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. Well-documented cases studies from other organizations doing similar kinds of work can also prove very useful so long as their differences are identified.

2.3.2 Bottom-up/Activity-Based Costing

The bottom-up cost estimation approach begins with the lowest level cost component and rolls it up to the highest level for its estimate. This method produces the most accurate estimates of cost but also requires the most data and is the most labor-intensive to create. A bottom-up estimate of a system's cost is created using costs reported from lower level components.

Lower level estimates are typically provided by the people who will be responsible for doing the work. This work is usually represented in the form of a Work Breakdown Structure (WBS), which makes this estimate easily justifiable because of its close relationship to the activities required by the project elements. This can translate to a fairly accurate estimate at the lower level. The disadvantages are that this process can place additional burden on workers and is typically not uniform across entities. In addition, every level may be victim to a layer of conservative management reserve which can result in an over estimate. The approach also requires detailed cost and effort data from throughout the system, so the method cannot be used early in the development cycle.

2.3.3 Expert Opinion

The expert opinion method simply involves querying experts in a specific domain and taking their subjective opinion as an input. The obvious drawback to this technique is that the estimate is only as good as the experts' opinions, which can vary greatly from person to person. Expert opinion is not always included as a scientifically valid estimation method because estimates generated using only expert opinion are the most difficult to justify and are typically only used when no other methods are available.

The benefits of this method are that experts can provide a quick estimate with minimal investment in the absence of empirical data. They can also account for other variables, such as customer demands or technology availability that other approaches may overlook. Unfortunately, many years of experience does not always translate into the right expertise. Moreover, since this technique relies on human judgment, it has low reliability because even the most highly competent experts can be wrong.

Expert opinion is most useful for confirming and informing other cost estimation methods. For example, parametric models are often calibrated using a combination of expert opinion and historical data. The analogy method is most effective when an expert determines how best to map one system to another. The bottom-up approach depends on experts to conduct low-level analyses of cost. A common technique for capturing expert opinion is the Delphi method which was improved and renamed Wideband Delphi (Dalkey 1969; Boehm 1981). These methods reduce natural human bias, improving the usefulness of data collected from experts.

2.3.4 Heuristics

Heuristic reasoning has been commonly used by engineers to arrive at quick answers to technical problems. Practicing engineers, through education, experience, and examples, accumulate a considerable body of contextual information. These experiences evolve into instinct or common sense that is seldom recorded. These can be considered insights, lessons learned, common sense, or rules of thumb, that are brought to bear in certain situations. In more precise terms, heuristics are strategies using readily accessible, though loosely applicable, information to control problem-solving in human beings and machines. Heuristics are common in psychology, philosophy, law, and engineering. Systems engineering cost estimation heuristics and rules of thumb have been developed by researchers and practitioners (Rechtin 1991; Boehm, Abts et al. 2000; Honour 2002) as shortcuts for decision making.

Ultimately, heuristics are based on experience and often provides valuable results. However, they face the same shortfalls as expert opinion: heuristics based on past experiences may not accurately describe changing environments and heuristics are only as good as the experiences upon which they are built. As with expert opinion, heuristics are best used in combination with other cost estimation techniques.

2.3.5 Top Down and Design to Cost

The top down or design to cost (DTC) technique is most typically used when budget restrictions on a system are pre-defined and non-negotiable. It can be useful when a certain cost target must be reached regardless of the technical features. However, the approach can often miss the low level nuances that can emerge in large systems. It also lacks detailed breakdown of the subcomponents that make up the system. It is up to managers and executives to constantly ensure that standards or targets for cost set early during development are not exceeded.

In the defense acquisition community, the DTC philosophy is used to set cost targets and to make program managers more cost-conscious early in the acquisition life cycle. The method can also encompasses the use of incentives and/or awards to encourage achievement of specific production or operation and support (O&S) cost goals (Gille 1988).

2.3.6 Parametric

The parametric cost estimation approach is the most sophisticated and most difficult to develop. Parametric models generate cost estimates based on mathematical relationships between independent variables (e.g., aircraft weight) and dependent variables (e.g., cost of materials). 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 at the heart of parametric modeling. These relationships are known as Cost Estimating Relationships (CERs) and are usually based upon statistical analyses of large amounts of data. Regression models are used to validate the CERs and operationalize them in linear or nonlinear equations. Developing CERs requires a detailed understanding of the factors that affect the phenomenon being modeled, the assumptions of the model in use, and the units of measure provided by the model.

The main advantage of using parametric models is that, once validated, they are fast and easy to use. Parametric models do not require as much information as other methods, such as activity-based costing and estimation by analogy, and can provide fairly accurate estimates. Parametric models can also be tailored to a specific organization's CERs. However, some disadvantages of parametric models are that they are difficult and time consuming to develop and require a significant amount of clean, complete, and uncorrelated data to be properly validated.

Although many parametric models are referred to as cost models, they are actually effort models since they are designed to provide an estimate of the human effort required to successfully deliver a system. In the United States, the person-month unit is equivalent to 152 person-hours as shown by the following logic. In one year there are 52 available work weeks. Subtract two weeks for vacation, two weeks for holidays, one week for sick leave, and one week for training. This leaves 46 weeks of available work. Assuming 40 hours per week, this results in:

$$\frac{(46\,weeks/year)\times(40\,hours/week)}{(12\,months/year)} = 153\,hours/month$$

Figure 2. Calculation of person-months

Rounded down to the nearest even number to make calculations easier and to capture the fact there are other reasons – such as travel – that a person may not be able to work, the number that is typically used is 152 hours. For some countries in Europe that follow a shorter work week, the number of hours per person-month is 138, which means they assume that there are 36 hours of available work time each week.

3 PROBLEM FORMULATION: A CASE STUDY OF THE F119 ENGINE

3.1 History and Practice of HSI

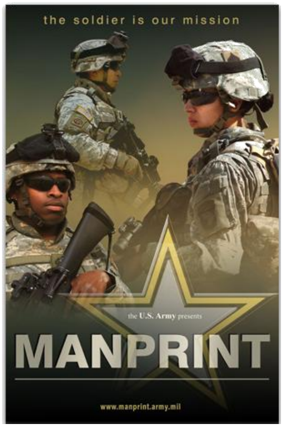
The study of human performance can be traced to at least as far back as the industrial revolution, when technological advances and a need for greater efficiency drove research on how humans could best interact with machines. At the time, these efforts were known as Industrial Engineering. The challenges and requirements of industry leading up to the beginning of the 20th century grew significantly during the first and second World Wars. In response, the U.S. and UK both funded efforts to understand human impacts on performance. Modern work in Human Factors derives from the research done during this time period (Nemeth 2004).

HSI has its origins in the field of Human Factors, with which it is commonly confused. Human Factors is "the study of those variables that influence the efficiency with which the human performer can interact with the inanimate components of a system to accomplish the system goals" (Proctor and Van Zandt 2008). Human Factors is also often understood to mean "the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data, and other methods to design in order to optimize human well-being and overall system performance" (International Ergonomics Association 2010). This second definition emphasizes the role of Human Factors in system performance and so overlaps with the definition of HSI (see section 1.1).

Human Factors is the field that HSI grew from and continues to be one of its central elements. However, HSI as it is practiced expands upon Human Factors by incorporating a broader range of human considerations over the system life cycle.

3.1.1 HSI in Defense Acquisition

General Maxwell R. Thurman of the U.S. Army is credited with first recognizing the need to integrate Human Factors Engineering (HFE) with other human domains early in the weapons system design process. In 1982, General Thurman directed that the Army's Human Factors program be expanded to include Manpower, Personnel Capabilities, and Training issues. The result was the U.S. Army Manpower and Personnel Integration Program (MANPRINT), established in 1984; it continues to define the Army's HSI policy today (Booher 2003).



Army MANPRINT Established 1984 7 Domains Deputy Chief of Staff, Army G1	Navy HSI Branch Formalized 1997 8 Domains CNO (N15)/ System Commands	Air Force HSI Office Formalized 2007 9 Domains Secretary of the Air Force for Acquisition (SAF/AQ)

Figure 6. Comparison of U.S. Military HSI programs (US Army MANPRINT Directorate 2007; Air Force Human Systems Integration Office 2009a; Naval Sea Systems Command 2009).

Although the Army's MANPRINT program has existed since the early 1980s, HSI as a field continues to mature. Figure 6 summarizes the HSI programs of the U.S. military branches (the Marine Corps is represented within the Navy's HSI program). Due to HSI's multidisciplinary nature, its stakeholders within each of the branches span departments and hierarchical structure. The HSI programs in each of the services are responsible for policy guidance and for assessment of programs, but specific design and analysis efforts in each of the domains is contracted to military assets or private firms possessing those capabilities. Therefore, the differences in organization of each of the HSI programs is not an indication of less emphasis put on a particular domain, but rather reflect the differences in each branch's existing practices.

Aside from domain differences, the HSI programs in each of the military branches fit into their larger organizational structures differently. The Army's MANPRINT program is part of Army G-1, the Deputy Chief of Staff responsible for Manpower and Personnel. The "Director, Training and Education Division (OPNAV (N15)) serves as the Navy's HSI and human performance advocate, and the Navy's single governance authority for HSI policy" (Chief of Naval Operations 2009). The Navy is unique in that its systems are acquired by each of the Navy Systems Commands, which report to the Secretary of the Navy. Each system command therefore also has its own HSI requirements division. The Air Force mandates addressing HSI concerns in all capabilities-based development documents in Air Force Instruction 10-601. The Air Force defines HSI as "a comprehensive management and technical approach for addressing the human element in weapon system development and acquisition" (Department of the Air Force 2006). The Air Force HSI Office serves as the policy arm of Air Force HSI and is currently part of the Secretary of Air Force for Acquisition (SAF/AQ). The "promotion, guidance, consultation, and implementation of HSI" in the Air Forces is the responsibility of the

Human Performance Integration Directorate, 711th Human Performance Wing (Department of the Air Force 2010).

The domains of HSI and their definitions recognized by INCOSE and adopted by the Air Force are shown in Table 1.

Table 1. Domains of human systems integration (International Council on Systems Engineering 2006).

Manpower	The number and mix of personnel (military, civilian, and contractor) authorized and available to train, operate, maintain, and support each system.
Personnel	The human aptitudes, skills, and knowledge, experience levels, and abilities required to operate, maintain, and support a system at the time it is fielded.
Training	The instruction and resources required providing personnel with requisite knowledge, skills, and abilities to properly operate, maintain, and support a system.
Environment	In the context of HSI, environment includes the conditions in and around the system and the concepts of operation that affect the human's ability to function as a part of the system as well as the requirements necessary to protect the system from the environment (e.g., radiation, temperature, acceleration forces, all-weather ops, day-night ops, laser exposure, air quality within and around the system, etc.).
Safety	The application of systems engineering and systems management in conducting hazard, safety and risk analysis in system design and development to ensure that all systems, subsystems, and their interfaces operate effectively, without sustaining failures or jeopardizing the safety and health of operators, maintainers and the system mission.
Occupational Health	The consideration of design features that minimize risk of injury, acute and/or chronic illness, or disability, and/or reduce job performance of personnel who operate, maintain, or support the system.
Habitability	Factors of living and working conditions that are necessary to sustain the morale, safety, health, and comfort of the user population that contribute directly to personnel effectiveness and mission accomplishment, and often preclude recruitment and retention problems.
Survivability	The ability of a system, including its operators, maintainers and sustainers to withstand the risk of damage, injury, loss of mission capability or destruction.
Human Factors Engineering	The comprehensive integration of human capabilities and limitations (cognitive, physical, sensory, and team dynamic) into systems design, to optimize human interfaces to facilitate human performance in training operation, maintenance, support and sustainment of a system."

3.1.2 HSI Best Practices

In 2003, the *Handbook of Human Systems Integration* combined many of the lessons learned from Hal Booher's 1997 case studies on Army Human Factors Integration (HFI) with the experience of other researchers in the field. The result was a set of ten "principles" described as "crucial to effective HSI" (Booher 2003). These principles are show in Table 2.

Landsburg et al. (2008) performed their own case studies on mostly non-military examples of HSI from the Department of Transportation, the Federal Aviation Administration and the U.S. Coast Guard. They derived an 11-step "guide" to HSI best practice, based on the U.S. Navy's HSI practices. They also created a prioritized list of elements critical to HSI success, summarized in Table 2. Landsburg et al. concluded that the transportation organizations studied would have benefitted from the implementation of a top-level HSI program modeled after the Navy's HSI program.

Booher (2003) consolidated detailed analyses of complex Army systems to create a direct link between HFI investment and cost savings. Landsburg et al. (2008) chose instead to focus on a few isolated HSI successes and then develop recommendations from the practice of HSI in Navy acquisitions.

Table 2. Contributors to HSI success.

The 10 Principles of Effective HSI (Booher 2003)	**Prioritized List of Critical Elements for Successful HSI (Landsburg, Avery et al. 2008)**
Top-level leadership	Management and Organizational Commitment
Focus on human-centered design (HCD)	User/stakeholder involvement
Source selection policy	Education and awareness of all
Organizational integration of all HSI domains	HSI process ownership
Documentation integration into procurement process	Holistic, enabled view
Quantification of human parameters	Funding support
HSI technology	Documented and technically sound processes
Test and evaluation/assessments	Qualified personnel
Highly qualified practitioners	Open collaborative environment
Education and training program	Practical applications based on sound Human Factors research

3.1.3 HSI in the context of Systems Engineering

Systems engineering standards have long recognized the role of the human in the system. Two approaches toward the integration of humans and systems currently exist in the literature. There is the argument that systems engineering "does not focus on the human component" –instead these issues are "the domain of the Human Factors specialist" (Proctor and Van Zandt 2008). However, recent policies on systems engineering tend to incorporate human considerations into existing practices. For example, IEEE Standard 1220, *Systems engineering — Application and* *management of the systems engineering process* states “complex components represent system elements that are composed of hardware, software, and/or humans” and “the human elements are integral to the systems hierarchy and may be present at any level” (International Organization for Standardization 2005). The INCOSE Systems Engineering Handbook states: HSI is “an essential enabler to systems engineering practice as it promotes a “total system” approach which includes humans, technology (hardware, software), the operational context and the necessary interfaces between and among the elements to make them all work in harmony.”

These documents do not explicitly highlight the differences between the practice of Human Factors and HSI. The distinction can be unclear and the roles of the two disciplines often overlap. Well-established policies and practices guide the work of Human Factors engineers: see, for example, (Proctor and Van Zandt 2008) and (Department of Defense 1989). Much less work has been done to isolate and identify the tasks or processes that make up HSI effort. Malone and Carson (2003) argue that “the primary objective of HSI in system acquisition is to influence design with requirements and constraints associated with human performance and accommodation” and suggest the following initiatives to achieve that objective:

- identify human performance issues and concerns early in system acquisition;
- define the roles of humans in system operations and maintenance early in system development;
- identify deficiencies and lessons learned in baseline comparison systems;
- apply simulation and prototyping early in system design to develop and assess HSI concepts;
- optimize system manning, training, safety, survivability, and quality of life;
- apply human-centered design; and
- apply human-centered test and evaluation.

While no one set of processes defines systems engineering, one set laid out in ANSI/EIA 632, *Processes for engineering a system* helps to illustrate the link between HSI and systems engineering (ANSI/EIA 1999). The work breakdown structure in Table 3 can be related to the higher level HSI initiatives above. For instance, the first initiative involving human performance issues and concerns early in the life cycle can be carried out by a number of detailed activities listed in Table 3: technical plans, system technical requirements, implementation, and transition to use. Parallels can be drawn between the recommendations of Malone and Carson and each of the fundamental processes of the systems engineering work breakdown structure – starting with acquisition and supply and continuing through test and evaluation. Likewise, the systems engineer should stay aware of HSI considerations throughout the entire system lifecycle.

Table 3. Systems engineering activities (ANSI/EIA 1999)

Fundamental Processes	**Process Categories**	**Activities**
Acquisition and Supply	Supply Process	(1) Product Supply
	Acquisition Process	(2) Product Acquisition, (3) Supplier Performance
Technical Management	Planning Process	(4) Process Implementation Strategy, (5) Technical Effort Definition, (6) Schedule and Organization, (7) Technical Plans, (8) Work Directives
	Assessment Process	(9) Progress Against Plans and Schedules, (10) Progress Against Requirements, (11) Technical Reviews
	Control Process	(12) Outcomes Management, (13) Information Dissemination
System Design	Requirements Definition Process	(14) Acquirer Requirements, (15) Other Stakeholder Requirements, (16) System Technical Requirements
	Solution Definition Process	(17) Logical Solution Representations, (18) Physical Solution Representations, (19) Specified Requirements
Product Realization	Implementation Process	(20) Implementation
	Transition to Use Process	(21) Transition to use
Technical Evaluation	Systems Analysis Process	(22) Effectiveness Analysis, (23) Tradeoff Analysis, (24) Risk Analysis
	Requirements Validation Process	(25) Requirement Statements Validation, (26) Acquirer Requirements, (27) Other Stakeholder Requirements, (28) System Technical Requirements, (29) Logical Solution Representations
	System Verification Process	(30) Design Solution Verification, (31) End Product Verification, (32) Enabling Product Readiness
	End Products Validation Process	(33) End products validation

3.1.4 HSI and Cost Estimation

Whereas this thesis asks what the "right" amount of HSI is for a system, previous work has largely focused on specific activities and related cost savings.

Harold Booher's 1997 *Human Factors Integration: Cost of and Performance Benefits to Army Systems* examines four Army systems and the impacts of Human Factors Integration (HFI), a term often used interchangeably with HSI. The case studies provide an assessment of costs that were avoided due to HFI considerations throughout the development process. Some costs were estimated using historical data on mishaps and occupational health impacts. Other data were generated using models that simulated the effects of system use on humans. At the time, the leading model was a software package called Hardware vs. Manpower, known by its shorthand of HARDMAN III (Booher 1997). HARDMAN III could assign specific tasks to simulated crewmembers and calculate the effort placed on each. The program could then make a recommendation as to the optimal crew size of a system. Today's incarnation of HARDMAN is the Improved Performance Research Integration Tool (IMPRINT), a model developed by the U.S. Army Research Laboratory for use across the DoD. The four case studies performed in 1997 showed how HFI and MANPRINT had improved Army systems and resulted in significant cost avoidance. The analysis focused on modeling techniques that were applied early in the development process and estimated costs avoided using historical data.

Booher's case studies are summarized in *Cost Arguments and Evidence for Human Factors Integration*, produced on behalf of the United Kingdom's Ministry of Defense (MoD) (Ministry of Defence 2006). The booklet explores the costs and benefits of HSI work, citing many specific examples. Sager and Grier (2005) also document a number of case studies from the domains of usability, training, and HFE as examples of the costs and benefits of doing HSI.

While most literature has focused on the costs and benefits or return-on-investment (ROI) of HSI, the work of this thesis focuses specifically on predicting necessary investment. Currently, budgeting for HSI in defense acquisition programs is assigned based on particular HSI activities expected to be performed during development. For example, budget may be set aside for iterative safety analyses or crewmember workload simulations, but budget is rarely designated for "HSI" in general. Varying sources have estimated different values of HSI investment as a fraction of development costs. Some heuristics include "from 0 to 8% of design costs" (Booher 1990) and "between 0.5% and 6% of developmental costs" (Hewitt 2003).

A review of common cost estimation approaches can be found in section 2.3. Sager and Grier (2005) suggest that two approaches can used to estimate Human Factors costs: "(1) By drawing analogies to similar situations, either via case studies or personal experience; and/or (2) by applying expert judgment to identify, for example, likelihood and impact of risk events." The UK MoD's *Cost-Benefit Analysis for Human Factors Integration: A Practical Guide* suggests three methods for estimating HFI (or HSI) costs:

1. As a percentage of project budget.
2. Breaking down the budget into components.
3. Parametric approach, based on number of studies needed (Bruseberg 2009).

Methods (1) and (2) apply a mixed heuristic- and expert-opinion-based approach: costs are estimated as a function of total development costs and weights are then applied by domain experts. Method (3) applies a parametric approach, but takes as an input "number of studies needed," which itself is determined by a combination of heuristics and expert opinion.

The Federal Aviation Administration's *Human Factors Assessments in Investment Analysis: Definition and Process Summary for Cost, Risk, and Benefit* takes a novel approach to estimating HSI effort. It suggests a number of "macroscopic cost drivers" that can be used early in a system's development to estimate Human Factors costs:

1. Definition of and Agreement on System Requirements.
2. The complexity of the human-system integration.
3. Organizational culture and nature of relationships among management, user, and provider unions, industry, and other stakeholders (e.g., interests converge or negotiations are necessary).
4. Pace of program (e.g., aggressive, normal, slow).
5. Safety and security considerations (e.g., higher security, or normal security).
6. Collaboration with international, external, or domestic organizations for standardization and other reasons (Hewitt 2003).

The cost estimation approaches discussed above suffer from two major shortfalls:

(1) They mostly pertain to the practice of Human Factors and do not give specific consideration of HSI. Existing cost estimation approaches rely on experts being able to either assess Human Factors risks and consequences or assess the number of Human Factors studies that will be required within system development. HSI encompasses many human-related domains. As shown in section 3.1.1, HSI domains can change by organization. It is therefore unlikely that any expert or group of experts would be able to accurately estimate a sufficient amount of HSI risks or studies needed to produce a credible cost estimate.

(2) They rely on the fact that heuristics or experts in HSI exist in the organizations where the estimation is to take place. Heuristics and expert opinion are certainly useful methods of cost estimation, but they should be used only when conditions support their application. The problem of finding credible HSI experts has just been discussed. Heuristics face a similar challenge to credibility. Heuristics are created through years of experience or are created using large data sets showing causal relationships. The problem of experience relates to the problem with identifying a true HSI "expert": very few people can be expected to possess the experience necessary to make effort predictions relevant to every domain of HSI. None of the cost estimation approaches above have performed sufficient analyses to establish a useful heuristic for HSI.

The next section describes a case study conducted to explore the research questions and gain insight into how cost estimation approaches for HSI could be developed.

3.2 Case Study

3.2.1 Case Study Methodology

This case study documents HSI activities during the development of Pratt & Whitney's F119 engine, which powers the $143 million Lockheed Martin F-22 Raptor fighter aircraft. The F-22 raptor fulfils the air superiority role in the Air Force by using a package of technologies to allow pilots to "track, identify, shoot and kill air-to-air threats before being detected" (Department of the Air Force 2009). Although the Air Force HSI Office was not formalized until 2007, much of the work done on the F-22 and F119 in the 1980s and 1990s spans the domains of HSI, making the F119 a best practice of HSI in the Air Force.

The design of the study was based on Yin's (2009) approach for identifying five important components to case study design: (1) a study's questions; (2) its proposition; (3) its units of analysis; (4) the logic linking the data to the propositions; and (5) the criteria for interpreting the findings.

<u>Study Questions</u>: The overarching goal of the case study was to document a best practice of HSI in the Air Force, with the hope that insights gained would inform subsequent research objectives. As a result, the case study would best be categorized as a descriptive case study.

The three research questions discussed in section 1.2.1 were used to guide the execution of the case study:

- R1: How can the "right" amount of effort to invest in HSI be determined?
- R2: How much does HSI effort cost?
- R3: What is the relationship between HSI and systems engineering?

The hypothesis of this thesis is that HSI effort can be measured as a function of systems engineering effort. The first approach to interpreting this hypothesis was that a quantitative heuristic could be identified relating HSI effort to systems engineering effort – for example, "HSI cost should always be between 20-25% of systems engineering costs." This case study sought to isolate HSI costs from systems engineering costs in order to establish such a relationship. The following proposition was therefore developed:

<u>Proposition</u>: HSI effort can be isolated from the larger systems engineering effort spent. If a quantitative relationship between HSI cost and systems engineering cost could be documented, it would represent a data point useful in the development of a cost model.

<u>Units of Analysis</u>: The unit of analysis was the development of the F119, from concept development until major engineering and manufacturing development (EMD). The case study focused primarily on work done by Pratt & Whitney, though it became apparent during the case study that interaction with the Air Force was also of importance.

<u>Logic Linking Data to Propositions</u>: No historical data on specific costs associated with HSI activities were available either because data were not kept or the records could not be found. Instead, the case study depended on Pratt & Whitney employees familiar with the F119 to build an understanding of its development. Interviews were conducted with Pratt & Whitney

engineers who were active in the development of the F119, in both technical and management roles. Interviews were also conducted with Air Force personnel familiar with the development and maintenance of the F119. Interviews were supplanted with existing literature.

Criteria for Interpreting Findings: The findings of descriptive case studies can be difficult to present and interpret. What data is available may be too sparse to draw statistically significant conclusions from. Yin (2009) recommends identifying rival propositions and linking evidence gathered to one or the other. The rival propositions in this case study are:

> Proposition: HSI effort could be isolated from the larger systems engineering effort spent.
> Rival: HSI effort could not be isolated from the larger systems engineering effort spent.

These propositions are addressed again at the conclusion of the case study.

3.2.2 Early Air Force Emphasis on Reliability and Maintainability

The Defense Resources Board approved the creation of the Advanced Tactical Fighter (ATF) program in November of 1981 to create a military jet that would be able to guarantee air superiority against the Soviet Union. This fighter was meant to replace the F-15 Eagle, which had previously filled this role. A team composed of Lockheed, Boeing, and General Dynamics competed against Northrop Grumman to develop the fighter. In 1991, the ATF contract was awarded to the Lockheed team's F-22, powered by Pratt & Whitney's F119 engine (Figure 7). Then Secretary of the Air Force Donald Rice noted that an important consideration in the awarding of the contract was the fact that the F-22's engines offered superior reliability and maintainability (Bolkcom 2007).

Figure 7. PW F119 engine cutaway (Pratt and Whitney 2002).

The Air Force placed an emphasis on reliability and maintainability from the beginning of the ATF program as well as throughout the Joint Advanced Fighter Engine program (JAFE) – the program to develop the engine for the ATF. In June of 1983, four general officers representing the Army, Navy, and Air Force signed a joint agreement in order to "emphasize to the DoD and defense contractor communities the critical importance of improving operational system availability by making weapon system readiness and support enhancement high priority areas for all our research and development activities" (Keith, Williams et al. 1983). Later that year, the director of the JAFE program sent a memorandum to participants in the program, including Pratt & Whitney, asking them to consider that over 50 percent of Air Force budget was then devoted to logistics, and that the problem would only worsen (Reynolds 1983).

To address this increase in logistics cost and determine ways to develop creative solutions, the Air Force created the Reliability, Maintainability & Sustainability (RM&S) program in 1984 (Gillette 1994). Besides reducing life cycle cost, the RM&S program also sought to address the reliability and durability problems that had plagued Pratt & Whitney's previous F100 engine, which powered the Air Force's F-15 Eagle. Developed in the 1970s, the F-15 was developed specifically to counter the Russian MiG-25. Therefore, emphasis was placed on performance during the development of both the F-15 and F100. Unfortunately, the high performance of the F100 meant that the engine was more prone to failure and downtime. By the 1980s, the Russian air superiority threat was no longer as pressing as when the F-15 was developed and supportability was emphasized over performance. As a result, the Air Force wanted improved RM&S not only on the F119 engine, but on development of the F-22 as a whole. Specific supportability goals for the F-22 were announced as early as 1983 (Aronstein, Hirschberg et al. 1998).

3.2.3 Understanding Customer Needs

The F-22 engine competition was not the only instance in which Pratt & Whitney had competed with General Electric. Both companies had developed engines to power the Air Force's F-16 Fighting Falcon. In the end, GE provided the majority of engines for that platform. Pratt & Whitney saw success in the JAFE program as critical to the company's ability to continue to compete in the military engine market. For the F119 engine, Pratt & Whitney decided not only to meet the Air Force's RM&S requirements, but to emphasize designing for the maintainer throughout all aspects of the program. The company's approach exemplified the best practices of what is now known as HSI.

Pratt & Whitney conducted approximately 200 trade studies as contracted deliverables for the Air Force. Pratt & Whitney engineers also estimated they had conducted thousands of informal trade studies for internal use. These trade studies used evaluation criteria, including safety; supportability; reliability; maintainability; operability; stability; and manpower, personnel, and training (Deskin and Yankel 2002).

Figures of merit were developed for the trade studies to define a consistent set of criteria upon which to assess the trade studies. Pratt & Whitney engineers used these figures of merit to determine which engineering groups would participate in each trade study.

As is often the case in the development of complex defense systems, responsibilities for the various domains of HSI were distributed among many different organizations at Pratt & Whitney. Of the nine domains of HSI (see Table 1 in section 3.1.1), seven were represented in Pratt & Whitney's engineering groups. Maintainability, Survivability, Safety, Training, and Materials were all engineering groups at Pratt & Whitney. Manpower, Personnel, and HFE were taken into account by the Maintainability group. HFE also impacted the Safety group. Occupational Health was considered by both the Safety group and Materials group, which dealt with hazardous materials as one of its responsibilities. While there was an Environmental Health and Safety

(EH&S) group at Pratt & Whitney, it dealt with EH&S within the organization itself and did not impact engine design. Habitability was not an important consideration in the engine design.

3.2.4 Top-Level Leadership and Integrated Product Development

The major requirements for RM&S came directly from the Air Force. The JAFE program in particular was intended to improve RM&S by "reducing the parts count, eliminating maintenance nuisances such as safety wire, reducing special-use tools, using common fasteners, improving durability, improving diagnostics, etc" (Aronstein, Hirschberg et al. 1998). While General Electric made significant RM&S improvements to its F120 engine during this time period, Pratt & Whitney centered its competitive strategy on RM&S superiority.

During the Joint Advanced Fighter Engine competition, Pratt & Whitney participated in the Air Force's "Blue Two" program. The name refers to the involvement of maintenance workers in the Air Force – "blue-suiters". The program brought Pratt & Whitney engineers to Air Force maintenance facilities so that the engine designers could experience first-hand the challenges created for maintainers by their designs. Maintainers showed how tools were poorly designed, manuals had unclear instructions, and jobs supposedly meant for one person took two or more to complete safely.

Many of the features for which the F119 would come to be praised were a result of leadership commitment to HSI. Frank Gillette, the Chief Engineer of the F119, served in various leadership positions on the F119 project, eventually leading a team of over 900 engineers. In interviews with Pratt & Whitney employees familiar with the F119, Gillette was identified as a driving force behind ensuring buy-in to HSI principles.

When the Pratt & Whitney team returned from its Blue Two experience to work on the F119, Gillette captured the lessons learned from the site visits in a series of presentations. These presentations were then shown to every engineer on the F119 team. Gillette also established design ground rules based on the requirements of the maintainer.

One of the most important requirements for the F119 was that only five hand tools should be used to service the entire engine. All Line Replaceable Units (LRUs) would have to be "one-deep", meaning that the engine would have to be serviceable without removal of any other LRUs, and each LRU would have to be removable using a single tool within a 20-minute window (Gillette 1994). Maintenance would have to be possible while wearing hazardous environment protection clothing. Maintenance tasks would have to accommodate the heights of maintainers from the 5th percentile female to the 95th percentile male. In addition:

> *"Built-in test and diagnostics were integrated with the aircraft support system, eliminating the need for a special engine support system. Lockwire was eliminated, and torque wrenches were no longer required for "B" nut installations. The engine was designed with built-in threadless borescope ports, axially split cases, oil sight gauges, and integrated diagnostics. Other improvements were a modular design…, color-coded harnesses, interchangeable components, quick disconnects, automated integrated maintenance system, no component rigging, no trim required, computer-based training, electronic technical orders, and foreign object damage and corrosion resistant. These advances were intended to reduce operational level and intermediate level maintenance items by 75% and depot level tools by 60%, with a 40% reduction in average tool weight"* (Aronstein, Hirschberg et al. 1998).

These innovations were only possible using the Integrated Product Development (IPD) concept. Whereas on previous projects, engineering groups at Pratt & Whitney each worked in their own respective disciplines, under IPD, teams of engineers from varying disciplines were able to provide design engineers with the perspectives they needed to see the full impacts of their design decisions.

3.2.5 Continuing Accountability and Enforcement of HSI

Adoption of the IPD concept brought various stakeholders together early in the design process and ensured multidisciplinary input through design and development. As a matter of policy, whenever a design change needed to be made, the originating group would submit the change to be reviewed by a Configuration Control Board (CCB). CCBs were composed of senior engineers from multiple engineering groups. At CCB meetings, each group with a stake in a particular design change would explain the impacts of that change to the chair of the CCB, typically a design engineer. The chair would then weigh the different considerations of the design change and either approve/disapprove the change or recommend further analysis be done.

In instances when Air Force requirements needed to be changed, the originating group would submit a Component Integration Change Request (CICR), which would then be internally debated much as with design changes. CICRs were typically initiated when it was determined that a particular requirement might not be in the best interests of the customer or when one requirement conflicted with another. Once a CICR was finalized internally by all of Pratt & Whitney's engineering groups, it was presented to the Air Force, which would then make the final decision on whether a requirement could be eliminated, modified, or waived.

The processes for design and requirement change ensured that the work of one group did not create unforeseen problems for another. However, change requests were typically made in response to problems that arose during development. Although reacting to and fixing these problems were important, it took proactive leadership to make sure HSI principles were being followed even when no problems were apparent.

Frank Gillette created several policies that ensured engineers kept RM&S considerations constantly in mind. All part design drawings were required to be annotated with the tools needed to service that part. This helped to achieve the goal of being able to ser-vice the entire engine with only five hand tools (in the end, the F119 required five two-sided hand tools and one other tool, sometimes described as 11 tools total).

Gillette also insisted on the development of several full-scale mock-ups of the F119. These mock-ups came at a considerable cost (over $2 million each, while the cost of an engine was then about $7 million) but allowed engineers to see whether their designs had really achieved maintainability goals. Engineers were asked to service LRUs on the mock-ups by hand to ensure that they were each indeed only "one-deep". When an LRU was shown to not meet that requirement, the teams responsible for those LRUs were asked to redesign them.

3.2.6 HSI Efforts Contribute to Competition Success

Leading up to the major EMD contracts awarded in 1991, Pratt & Whitney conducted 400 distinct demonstrations of the F119's RM&S features. The F119 also accrued over 110,000 hours of component tests and 3,000 hours of full-up engine tests, representing a thirtyfold increase in total test hours over its predecessor, the F100 (Aronstein, Hirschberg et al. 1998). Pratt & Whitney was willing to spend significant effort on demonstrating the F119's RM&S features because the company had recently been beat out by GE in their competition to provide engines for the Air Force's F-16 Fighting Falcon and therefore saw the Joint Advanced Fighter Engine competition as its last chance to stay in the military engine market.

In 1991, both Pratt & Whitney and General Electric were awarded contracts worth $290 million to complete the EMD phase of competition. The companies were given independence as to the number and types of tests that would be run on their engines, while the Air Force provided safety oversight. As a result, Pratt & Whitney chose to log about 50 percent more test hours than General Electric (Aronstein, Hirschberg et al. 1998).

GE chose to emphasize the performance of its F120 engine over RM&S, though the F120 did meet the Air Force's RM&S requirements. The F120 was the world's first flyable variable cycle engine (Hasselrot and Montgomerie 2005). This meant that the F120 was able to change from turbofan to turbojet configuration to achieve maximum performance in multiple flight situations. The F120 was tested in both Lockheed's YF-22 and Northrop Grumman's YF-23 prototypes, demonstrating better maximum speed and supercruise than Pratt & Whitney's F119 in both cases(Aronstein, Hirschberg et al. 1998). The dry weight of the F119 is classified, making it impossible to calculate its exact thrust-to-weight ratio. However, Pratt & Whitney advertises the F119 as a 35,000 lb thrust class engine, putting it into the same thrust class as the F120 (Gunston 2007).

Despite the F120's superior performance in the air and higher thrust-to-weight ratio, on April 23, 1991, the Air Force chose the combination of Pratt & Whitney's F119 and Lockheed's YF-22 to be developed into the F-22. Pratt & Whitney had repeatedly demonstrated a better understanding of the Air Force's RM&S needs, investing more time and money into demonstrations and internal efforts than its competitor. It also avoided the increased risk of developing a variable cycle engine, at the time considered a relatively new and untested technology. By 1991, the Air Force's RM&S program was less focused on reducing downtime and more concerned with reducing life cycle costs. Pratt & Whitney had presented a management plan and development schedule that the Air Force considered sensitive to their needs (Aronstein, Hirschberg et al. 1998). On August 2, 1991, contracts worth $11 billion were awarded to Lockheed and Pratt & Whitney (Bolkcom 2007) demonstrating the Air Force's commitment to HSI. Pratt & Whitney's portion was worth $1.375 billion alone (Aronstein, Hirschberg et al. 1998).

3.2.7 Key HSI Success Factors

The Air Force's early and continuing emphasis on RM&S was captured via requirements. Although dating back to 2003 the General Accounting Office (GAO, now the Government Accountability Office) was still advocating for more equal consideration of reliability and maintainability in requirements definition (General Accounting Office 2003), this case study showed that the Air Force had already understood this principle a decade prior. The Air Force's initial guidance to emphasize RM&S shaped the design approach of all of its contractors.

The actions of both the Air Force and Pratt & Whitney were examples of combining top-level leadership's role with sound systems engineering practices. From a systems engineering standpoint, the Air Force set formal requirements and expected deliverable trade studies based on HSI concerns. In terms of leadership, the Air Force set early supportability goals, distributed memoranda explaining their intent, and funded programs to show Pratt & Whitney engineers actual maintenance conditions. For systems engineering, Pratt & Whitney embraced the IPD approach along with IPD's subordinate systems engineering processes. The company made sure to include diverse engineering groups on all major design and configuration changes, a practice it continues to today. In terms of leadership, Pratt & Whitney invested significant effort to develop mock-ups, conduct extra testing, and hold engineers accountable for RM&S standards, all of which led to HSI success. These combined efforts of customer and contractor to define clear requirements and communicate common expectations led to product success.

The efforts described above can be summarized into several key success factors:

1. Air Force policy to elevate visibility of HSI early in development.
2. Pratt & Whitney's adoption of the Integrated Product Development approach, which ensured engineering organizations responsible for each HSI domain had a voice.
3. The integration of HSI and systems engineering in the early phases of the acquisition life cycle.
4. Participation in the "Blue Two" program, which ensured Pratt & Whitney engineers understood the challenges facing actually maintainers.

3.2.8 Conclusions

Conclusions are drawn by addressing the research questions identified for this case study and then applying those insights to the case study proposition.

Research questions and insights:

1. How did Pratt & Whitney determine how much HSI effort would be needed?

Pratt & Whitney performed over 200 deliverable trade studies for the Air Force and thousands of internal trade early in development. These trades considered many of the domains of HSI and integrated those costs into the system.

2. How much did HSI effort eventually cost?

Some specific costs associated with HSI were identified. For example, the “Blue Two program”, additional engine mockups, and additional hours spent on test could all be associated with costs that improved HSI. However, there was no way for Pratt & Whitney to separate out the costs of HSI from the rest of development. HSI was integrated into too many parts of the engine to be accounted for in such a way

3. How did HSI fit into the larger systems engineering picture?

IPD drove systems engineering effort at Pratt & Whitney and HSI was an integral component of IPD. IPD brings together stakeholders from across Pratt & Whitney’s engineering organizations to make trades and decisions.

The propositions of the case study were defined as follows:

> Proposition: HSI effort could be isolated from the larger systems engineering effort spent.
> Rival: HSI effort could not be isolated from the larger systems engineering effort spent.

The evidence collected through interviews and literature during the case study have supported the rival proposition, that HSI effort could not be isolated from the large systems engineering effort spent.

3.2.9 Limitations

A descriptive case study was applied in this instance due to recommendations made by Yin (2009) relating to access to personnel and data. Although interviews were conducted with several engineers with detailed knowledge of the F119, the unit of analysis of the case study had ended in 1991, 17 years before the case study was begun. Memories likely faded in that time. Specific costs related to HSI may have been more readily available had the case study been conducted shortly earlier.

Definitions and perceptions of HSI from the time period of the unit of analysis differed from present-day policy, both at Pratt and Whitney, and within government. Not every current domain recognized by INCOSE was addressed during the development of the F119. However, the challenges faced in the execution of the case study reflect those that would impede any researcher interested in HSI. Acquisition projects often take years to complete, and the domains of HSI applied in any particular program shift in response to program priorities.

3.2.10 Takeaways/Next Steps

The overarching goal of the case study was to document a best a practice of HSI in the Air Force, with the hope that insights gained would inform subsequent research objectives. This section summarizes the insights drawn from the case study relevant to the rest of the research described in this thesis.

Emphasis on Requirements: It was shown that much of the work that led to HSI success resulted from formal deliverables required by the Air Force. The trade studies that were done early in development represent the exploration of how to fulfill requirements. Later in development, the process for recommending changes to requirements became a part of IPD/systems engineering and incorporated the perspectives of multiple HSI stakeholders. Going forward in this research, requirements will be examined as a driver of HSI effort.

Early Decisions Define the System: The unit of analysis of the case study was the development of the F119 from concept development through to the beginning of EMD. Previous literature has shown that engineers often think of human considerations as part of test and evaluation, occurring later in a system's life cycle (Harrison and Forster 2003). However, as shown in the case study, decisions made during early phases of development define what types of effort will be emphasized and predict the success or failure of HSI. Further research will therefore focus on these early stages of development.

Importance of Teams: IPD was identified as critical to HSI success. A defining characteristic of IPD is the use of integrated product teams (IPTs) to perform trades and collaborate on major decisions. IPTs have become a hallmark of sound systems engineering practice. Further research will therefore give consideration to the need to factor in multiple points of view when making effort predictions for HSI.

3.3 Introduction of HSI into a Systems Engineering Cost Model

The hypothesis of this thesis is:

- H1: Human Systems Integration effort can be estimated as a function of total Systems Engineering Effort.

The case study described in section 3.2 showed that no direct numerical relationship between HSI effort and systems engineering effort could be identified; HSI was integrated throughout systems engineering. Instead, insights gained from the case study highlighted the need to consider requirements when estimating HSI effort. As a result, the following sub-hypothesis was developed:

- H2: Human Systems Integration effort can be estimated by counting "number of HSI-related requirements."

Another major insight gained through the case study was that any HSI cost estimation would need to occur early in development, before requirements had been finalized. As a result, the Constructive Systems Engineering Cost Model (COSYSMO) was identified for further exploration.

3.3.1 The Constructive Systems Engineering Cost Model (COSYSMO)

The Constructive Systems Engineering Cost Model (COSYSMO) is a parametric model used to estimate systems engineering effort. The operation of parametric models is discussed in section 2.3.6. Parametric models are most often associated with cost estimation around the Milestone B decision point (Roper 2006; Defense Acquisition University 2009). However, they can be applied at any point prior to Milestone B, given the necessary inputs are available. It should be kept in mind, however, that estimates based off of changing or poorly defined inputs will be less precise than estimates with more defined inputs. COSYSMO uses systems engineering size drivers to produce its estimates. An example of how to use COSYSMO, including how to capture the inputs needed by the model, is given in section 4.3

3.3.1.1 Model Form

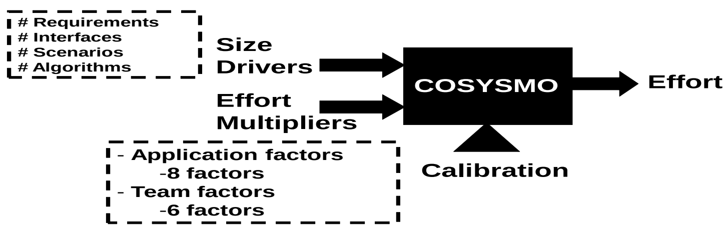


Figure 8. COSYSMO operational concept.

The operational concept for COSYSMO is illustrated in Figure 8. In order to use the model, estimators need to understand the expected technical capabilities of the system to be developed and make basic assumptions about the organization performing the technical work. COSYSMO requires no complex calculations on the part of the user. System characteristics are simply assigned complexity ratings such as “easy” or “difficult” and the appropriate effect on effort is calculated based on the Cost Estimating Relationship. However, COSYSMO does allow more advanced users to calibrate the model to their specific organizations in order to increase the model’s accuracy. The specific parameters to COSYSMO are described in the next section.

COSYSMO is a parametric cost model. As described in section 2.3.6, parametric cost models generate cost estimates based on mathematical relationships between independent variables (e.g., requirements) and dependent variables (e.g., effort). These relationships are known as Cost Estimating Relationships (CERs). The basic CER embedded in COSYSMO includes additive, multiplicative and exponential parameters as shown in Equation 1.

Equation 1. COSYSMO Cost Estimating Relationship (CER).

where:

PM	=	effort in person-months
A	=	calibration constant derived from historical project data
Size	=	determined by computing the weighted sum of the four size drivers
E	=	economy/diseconomy of scale; default is 1.0
n	=	number of cost drivers (14)
$\mathbf{EM}_{\boldsymbol{i}}$	=	effort multiplier for the i_{th} cost driver; nominal is 1.0.

The general rationale for whether a factor is additive, exponential, or multiplicative comes from the following criteria (Boehm, Valerdi et al 2005):

A factor is additive if it has a local effect on the included entity. For example, adding another source instruction, function point entity, requirement, module, interface, operational scenario, or algorithm to a system has mostly local additive effects. From the additive standpoint, the impact of adding a new item would be inversely proportional to its current size. For example, adding 1 requirement to a system with 10 requirements corresponds to a 10% increase in size while adding the same single requirement to a system with 100 requirements corresponds to a 1% increase in size.

A factor is multiplicative if it has a global effect across the overall system. For example, adding another level of service requirement, development site, or incompatible customer has mostly global multiplicative effects. Consider the effect of the factor on the effort associated with the product being developed. If the size of the product is doubled and the proportional effect of that factor is also doubled, then it is a multiplicative factor. For example, introducing a high security requirement to a system with 10 requirements would translate to a 40% increase in effort. Similarly, a high security requirement for a system with 100 requirements would also increase by 40%.

A factor that is exponential has both a global effect and an emergent effect for larger systems. 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 SoS integration, then it is treated as an exponential factor.

The size drivers and cost drivers of COSYSMO were determined via a Delphi exercise by a group of experts in the fields of systems engineering, software engineering, and cost estimation. The definitions for each of the drivers, while not final, attempt to cover those activities that have the greatest impact on estimated systems engineering effort and duration. These drivers are further discussed in the next two sections.

3.3.1.2 Size Drivers

It can be empirically shown that developing complex systems like a satellite ground station represents a larger systems engineering effort than developing simple systems, such as a toaster. In order to differentiate the two, four size drivers were developed to help quantify their relative complexities. The role of size drivers is to capture the functional size of the system from the systems engineering perspective. They represent a quantifiable characteristic that can be arrived at by objective measures.

Since the focus of COSYSMO is systems engineering effort, its size drivers need to apply to software, hardware, and systems containing both. They are: (1) *Number of System Requirements*, (2) *Number of System Interfaces*, (3) *Number of System-Specific Algorithms*, and (4) *Number of Operational Scenarios*. A more detailed discussion on the use of the *Number of Requirements* driver to estimate HSI effort is addressed in section 3.3.2.

3.3.1.3 Effort Multipliers

Table 4. Fourteen cost drivers and corresponding data items

Driver Name	**Data Item**
Requirements understanding	Subjective assessment of the understanding of system requirements
Architecture understanding	Subjective assessment of the understanding of the system architecture
Level of service requirements	Subjective difficulty of satisfying the key performance parameters (i.e., reliability, maintainability, manufacturability, etc.)
Migration complexity	Influence of legacy system (if applicable)
Technology risk	Maturity, readiness, and obsolescence of technology
Documentation to match life cycle needs	Breadth and depth of required documentation
# and Diversity of installations/platforms	Sites, installations, operating environment, and diverse platforms
# of Recursive levels in the design	Number of applicable levels of the Work Breakdown Structure
Stakeholder team cohesion	Subjective assessment of all stakeholders and their ability to work together effectively
Personnel/team capability	Subjective assessment of the team's intellectual capability
Personnel experience/continuity	Subjective assessment of staff experience in the domain and consistency on the project
Process capability	CMMI level or equivalent rating
Multi-site coordination	Location of stakeholders and coordination barriers
Tool support	Subjective assessment of SE tools

A group of fourteen effort multipliers have been identified as significant drivers of systems engineering effort. These are used to adjust the nominal person-month effort of the system under development. Each driver is defined by a set of rating levels and corresponding multiplier factors. The nominal level always has an effort multiplier of 1.0, which has no effect on the CER. Off-nominal ratings change the overall estimated effort based on pre-defined values.

Assigning ratings for these drivers is not as straight forward as the size drivers mentioned previously. The difference is that most of the cost drivers are qualitative in nature and require subjective assessment. A list of the fourteen cost drivers is provided in Table 4 with the corresponding data items or information needed in order to assess each driver.

If the ratings for effort multipliers associated with HSI effort are expected to vary significantly from ratings for systems engineering in general, the effort multipliers can be adjusted in relation to only the work being done by the HSI organization. An example is outlined in section 4.3. However, this approach assumes that HSI activities can be singled out from systems engineering effort, which, as previously discussed, can be difficult.

3.3.1.4 Number of System Requirements Size Driver

Of the four COSYSMO size drivers, *Number of Requirements* was selected for further research with regard to the role of HSI effort within systems engineering, based on the insights gained from literature review and the case study discussed in section 3.2.

HSI adds to the challenges in defining the *Number of Requirements* size driver. The COSYSMO definition of the *Number of Requirements* size driver is provided below.

> *"Number of System Requirements: This driver represents the number of requirements for the system-of-interest at a specific level of design. The quantity of requirements includes those related to the effort involved in system engineering the system interfaces, system specific algorithms, and operational scenarios. Requirements may be functional, performance, feature, or service-oriented in nature depending on the methodology used for specification. They may also be defined by the customer or contractor. Each requirement may have effort associated with it such as verification and validation, functional decomposition, functional allocation, etc. System requirements can typically be quantified by counting the number of applicable shalls/wills/shoulds/mays in the system or marketing specification. Note: some work is involved in decomposing requirements so that they may be counted at the appropriate system-of-interest* (Valerdi 2008).

As mentioned in the definition of the size driver, it must be assured that requirements have been decomposed to the correct level before being counted in COSYSMO. The counting rules adopted in COSYSMO are summarized here:

1. Determine the system of interest.
2. Decompose system objectives, capabilities, or measures of effectiveness into requirements that can be tested, verified, or designed.
3. Provide a graphical or narrative representation of the system of interest and how it relates to the rest of the system.
4. Count the number of requirements in the system/marketing specification or the verification test matrix for the level of design in which systems engineering is taking place in the desired system of interest.
5. Determine the volatility, complexity, and reuse of requirements. (Valerdi 2008)

Counting rules 1-4 are further explored in chapter 4, implementation and validation. They are the subject of significant further research in this thesis. The fifth counting rule is discussed here.

Many of the problems faced when trying to consistently count requirements for the purposes of cost estimation are more complicated when the requirements span multiple HSI domains. In addition, the definition of what constitutes an "HSI requirement" can vary between stakeholders, since different stakeholders count different domains under HSI. Stakeholders therefore often disagree on how to designate a requirement's complexity. This is due in part to the different types of requirements (i.e., functional, operational, environmental) that are used to define systems and their functions, the different levels of requirements decomposition used by organizations, and the varying degree of quality of requirements definition (how well they are written). The first four counting rules in COSYSMO help to mitigate many of these issues, but not all can be addressed. Therefore, complexity ratings can be assigned to individual requirements that affect their weight. The complexity ratings are summarized in Table 5.

Table 5. Number of system requirements rating scale (Valerdi 2008).

Easy	Medium	Difficult
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 requirements overlap

Further discussion of complexity ratings and how to assign them can be found in (Valerdi 2008).

This section introduced the COSYSMO model and the role of requirements in estimating systems engineering size. The next section explores the relationship between requirements engineering, systems engineering, and HSI with the goal of identifying areas for improvement.

3.3.2 Requirements as a Driver of Systems Engineering and HSI Effort

3.3.2.1 Role of Requirements in Systems Engineering and Acquisition

Requirements are central to the practice of systems engineering. Two widely-accepted definitions are:

> *"a statement that identifies a system, product or process' characteristic or constraint, which is unambiguous, clear, unique, consistent, stand-alone (not-grouped), and verifiable, and is deemed necessary for stakeholder acceptability"* (International Council on Systems Engineering 2006).

> *"a statement that identifies a product or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability (by consumers or internal quality assurance guidelines)"* (International Organization for Standardization 2005).

In the Department of Defense, stakeholder requirements are expressed as capabilities, defined as:

> *"The ability to achieve a desired effect under specified standards and conditions through combinations of means and ways across doctrine, organization, training, materiel, leadership and education, personnel, and facilities (DOTMLPF) to perform a set of tasks to execute a specified course of action. It is defined by an operational user and expressed in broad operational terms in the format of an Initial Capabilities Document or an Initial Capabilities Document (ICD) or a joint, DOTMLPF change recommendation (DCR). In the case of materiel proposals/documents, the definition will progressively evolve to DOTMLPF performance attributes identified in the Capability Development Document (CDD) and the Capability Production Document (CPD)"* (CJCS 2009).

The Department of Defense has shifted toward the use of the term "capabilities" rather than "requirements" in order to emphasize that a system or program does not always need to be built in order to achieve a desired outcome. Often, needs can be satisfied by changing the use of doctrine, organization, training, materiel, leadership and education, personnel, or facilities (DOTMLPF). However, once it is decided that the needed capabilities require a new program to be initiated, the resulting set of capabilities can be analyzed in the same way as requirements. Therefore, further use of the term "requirements" can refer to any of the three definitions listed above.

3.3.2.2 Requirements Engineering

Much of the work on requirements engineering has been in the realm of software engineering; a definition of requirements engineering adapted to systems engineering follows:

"Requirements engineering is the branch of engineering concerned with the real-world goals for, functions of, and constraints on systems. It is also concerned with the relationship of these factors to precise specifications of system behavior and to their evolution over time and across families of related systems" (Laplante 2009). As described in detail in (Hull, Jackson et al. 2005), "requirements engineering has a vital role to play at every stage of development." It is an integral part of systems engineering and a driver of systems engineering effort.

As systems become more complex and software-intensive, the practice of rigorous systems engineering becomes more critical to program performance. As discussed in section 2.2, the modern practice of systems engineering spans both technical and social disciplines. Likewise, requirements engineering is both an engineering and a humanistic endeavor, since understanding individual human behavior and social dynamics is critical to delivering systems that meet users' needs and expectations.

Effort spent on systems engineering has been shown to directly correlate to program schedule and performance (Elm, Goldenson et al. 2008). It follows, therefore, that the quality of requirements engineering should be of similar importance, as related work has supported (Hofmann and Lehner 2001; Kamata and Tamai 2007).

While previous studies have emphasized the importance of quality systems engineering and requirements engineering practice, there continues to be a disconnect between academic research and industry practice (Müller 2005). One study of software requirements showed that inspection of software requirements was mostly informal and ad hoc about one-quarter of the time (Neill and Laplante 2003).

One contributor to this problem is that academic research is difficult to validate without industry support. Empirical research in particular is lacking and any that does exist tends to be within the context of a classroom environment (Höfer and Tichy 2007; España, Condori-Fernandez et al. 2009).

3.3.2.3 Requirements Decomposition/Derivation

At the heart of requirements engineering lies requirements decomposition and derivation. The counting rules in COSYSMO asks the system stakeholder to either derive or decompose requirements to a level at which they may be input into the cost model. The counting rules are agnostic; the method of decomposition or derivation is not important, the counting rules are meant to be general enough to accommodate multiple methods.

When a requirement is decomposed, it is broken down into two or more "requirements whose total content is equal to the content of the original one," whereas a derived requirement is created from an existing requirement, but is different, and so the original requirement is not replaced (Eigner, Haesner et al. 2002). Although the distinction between decomposition and derivation is important, the end goal for the purposes of cost estimation is the same: to arrive at a set of requirements that can analyzed for the purposes of estimating systems engineering effort. For that reason, it should be assumed that future mention of "requirements decomposition" in this thesis includes both decomposition and derivation, for simplicity sake.

3.3.2.4 Functional and Nonfunctional Requirements

Whether a given requirement will result in decomposed or derived requirements depends largely on whether the original requirement is a functional or nonfunctional requirement.

A system specification may contain many different types of technical requirements varying in nature and complexity. Functional requirements are the fundamental or essential subject matter of the system. They describe what the product has to do or what processing actions it is to take. An example of a functional requirement is "The engine shall provide a thrust-to-weight ratio of T." Each functional requirement should have a criterion or use case. These serve as benchmarks to allow the systems engineer to determine whether the implemented product has met the requirement.

Functional requirements are more likely to be decomposed than derived. In Defense Acquisition, functional requirements go through the JCIDS process, which means they must be linked to Concepts of Operations and are reviewed at major milestones. They may be complex, but are unlikely to be replaced by a derived requirement. Instead, subsequent requirements would describe the original requirement, just in greater detail.

Nonfunctional requirements are the properties that the system must have, such as performance and usability. These requirements are as important as functional requirements to a product's success, but are not always weighted accordingly (GAO 2003b).

As previous work has shown, early systems engineering decisions make significant impacts on system life cycle costs (Bahill and Henderson 2005). Oftentimes, costs are driven by "nonfunctional" requirements, which are generally defined as requirements that must be met but are not central to defining a system's core function or capability (Neill and Laplante 2003). The understood definition of a nonfunctional requirement varies across organizations, but it is clear that all requirements, functional or not, are the responsibility of the systems engineer to realize (Glinz 2007).

Nonfunctional requirements are more likely to result in derived requirements because nonfunctional requirements do not go through the same checks as do functional requirements in JCIDS. Nonfunctional requirements are often added to requirements documents by acquisition professionals after draft functional requirements have been submitted. As a result, nonfunctional requirements will often contain generic language that must be adapted to the system of interest. Once new requirements pertinent to the system-of-interest have been derived, they too will need to be decomposed to a level appropriate to be counted in COSYSMO.

3.3.2.5 HSI and Requirements

HSI requirements are often expressed as nonfunctional requirements because they describe usability, operational and maintainability characteristics of the system. However, they can also be expressed as functional requirements, if a functional requirement pertains to HSI or one of its domains.

One challenge to counting of "HSI requirements" for the purposes of estimating HSI effort is that the term "HSI requirement" is not clearly defined. The Air Force HSI Office's *Human Systems Integration Requirements Pocket Guide* states "HSI practitioners should avoid thinking in terms of 'HSI Requirements' which suggest that there should be unique requirements specific to HSI. Instead, the HSI community should focus on the fact that any requirement may have HSI implications and that the role of HSI community is to highlight the human considerations that naturally occur as part of good and effective capability based requirements"(Air Force Human Systems Integration Office 2009b).

While the sentiments expressed in the Pocket Guide are sound, they leave a gap in semantic understanding. The term "HSI requirements" is used in a number of policies and documents (for example, in the Pocket Guide itself) and a definition helps to ensure stakeholders agree when speaking about HSI requirements.

The following definition for an "HSI requirement" is therefore proposed: "HSI requirements include, but are not limited to, any requirement pertaining to one or more domains of HSI, or the integration of those domains. Broadly, the term encompasses any requirement that contributes to the integration of human considerations into the system being developed." So as to reduce confusion in terms, "HSI-related requirements," "HSI-relevant requirements," and "HSI requirements" are all meant to represent the same concept in this thesis.

The term “nonfunctional” incorrectly implies that requirements that are not functional requirements must have a smaller impact on a system than functional requirements. Reliability, user interface, and safety are just some of the types of nonfunctional requirements that clearly defy such a simplification. Since HSI requirements are often nonfunctional requirements, they face the same stigma.

While the definition of nonfunctional requirements contributes to the difficulties of properly accounting for their impact on systems engineering effort, the difficulties are exacerbated by the requirements elicitation structure of the Department of Defense (DoD). In the United States, the DoD is the largest government customer of large systems acquisitions and is a leading agency in the evolution of systems engineering best practices and standards. The DoD defines system requirements using CDDs (see section 2.1.2). CDD’s assign a hierarchy of importance to requirements explicitly using a name scheme and implicitly, via their position within the CDD.

The next section explores how the challenges facing nonfunctional and particularly HSI requirements might be overcome.

4 IMPLEMENTATION AND VALIDATION

This section discusses the quantitative exploration of the hypotheses of this thesis. Two workshops were conducted and the insights gained from both are integrated into an example application of the COSYSMO model with implications for cost estimation in general.

4.1 Workshop 1: Application of Requirements Counting Rules

4.1.1 Overview

The Annual International Forum on the Constructive Cost Model (COCOMO) and Systems/Software Cost Modelling brings together representatives from industry, academia, and government interested in furthering the development of cost models such as COSYSMO. An exercise on requirements decomposition was developed in order to capitalize on the expertise of the Forum's participants.

The 2009 Forum was held at MIT from November 2nd to 5th, 2009. A workshop on COSYSMO was held as part of the program on November 5th. The research exercise described in the following section occurred between 10AM and 12PM during this workshop. Use of the term "workshop" refers only to work done during this period, not to the COSYSMO workshop as a whole.

4.1.2 Research Design

4.1.2.1 Research Question and Hypotheses

The research question explored in the workshop was developed from the existing hypothesis and sub-hypotheses of this thesis:

> H1: Human Systems Integration effort can be estimated as a function of total Systems Engineering Effort.
>
> > H2: Human Systems Integration effort can be estimated by counting "number of HSI-related requirements"
> >
> > > Workshop 1: Existing counting rules can be adapted to better account for HSI requirements

<u>Research Question</u>: How can COSYSMO counting rules be modified to improve counting of HSI-related requirements?

Two testable hypotheses were then developed from this research question:

<u>Hypothesis #1:</u> Using the cost estimation decomposition steps will produce requirements counts with high reliability across respondents.

<u>Hypothesis #2:</u> The cost estimation counting rules will help users quantify the number of HSI requirements to be input into COSYSMO.

4.1.2.2 Methods

Method Selection: The experimental simulation approach, as elucidated by McGrath (McGrath 1966) was used to develop the workshop. Experimental simulations are used when the experimenter can put participants into "contrived or created settings" (McGrath 1981). Laboratory experiments also put participants into created settings, but experimental simulations differ from laboratory experiments in that experimental simulations seek to "recreate or simulate the central features of some set of phenomenon which are of interest" (Klimoski 1978) whereas laboratory experiments "attempt to create a generic or universal 'setting'"(McGrath 1981).

The workshop sought to simulate how COSYSMO is used in real-world environments and identify possible improvements, particularly with respect to its application to HSI. In real-world systems, requirements are developed and refined by IPTs, groups of individuals whose personal views and relationships can affect the process. Context is also important – requirements take on differing meanings and difficulties, depending on the system being designed.

Experimental simulations facilitate negotiations, allowing the capture of group-related variables that would otherwise be difficult to control. They also provide participants with a realistic, as opposed to a generic context.

Procedures: Subjects were paired into teams of two to simulate the natural discussion that would occur in a requirements IPT. Teams were limited to two participants in order to maximize the number of useful data points generated.

Participants were asked to imagine they were participating on a requirements integrated product team responsible for the design of a "glass console" that would replace a standard SUV console, modeled off of the glass displays that have replaced traditional displays in some airliner cockpits. The system boundary was defined as shown in Figure 9.

Figure 9. SUV console design task (redjar 2007)

The format and content of the displays was modeled after requirements taken from government-furnished documents. An example is shown in Figure 10.

Cautions and Warnings. Method for displaying system warnings, cautions, and alarms must be appropriate given the importance of the situation (**Threshold**).

Figure 10. Sample requirement used during workshop 1.

The workshop adhered to the following schedule:

1. Distribute data collection forms (see Appendix 7.1)
2. Background information
    a. Purpose and goals of workshop
    b. Counting rules for cost estimation
    c. Definition of HSI
3. Introduce Task 1
4. Participants complete survey task 1
    a. Participants are given 2 minutes to analyze one requirement
    b. Participants record data for eight requirements.
5. Participants repeat process for survey tasks 2 and 3.

<u>Survey Tasks</u>: Three of the five counting rules for COSYSMO were developed into survey questions. Changing the counting rules into questions allowed quantitative data to be collected quickly and analyzed uniformly. The adaptations were made based on the full descriptions of the COSYSMO counting rules from (Valerdi 2008).

The adaptations made to counting rules 1 and 2 are straightforward and are summarized in Figure 11. The adaptation made in counting rule 3 warrants further explanation. The description of rule 3 of the COSYSMO counting rules is as follows:

> *"This step focuses on the hierarchical relationship between the system elements. This information can help describe the size of the system and its levels of design. It serves as a sanity check for the previous two steps"* (Valerdi 2008).

As discussed in section 3.3.2.4, HSI requirements are often represented as nonfunctional requirements. If proper systems engineering is done, nonfunctional requirements will be mapped across a systems specification to the areas they affect. In fact, the original COSYSMO guideline states that rule 3 should act as a "sanity check." Therefore, the adaptation of the rule into a survey question adopted the "sanity check" mentality by asking participants to estimate only whether a given HSI requirement would decompose to zero, one, or many requirements.

Some HSI requirements might pertain to higher or lower level systems and so would not affect the requirements count of the system of interest. Others might be written and counted the same way as any other requirement, but contain some element relevant to HSI. Still others might result in many derived requirements (a description of derived vs. decomposed requirements can be found in section 3.3.2.3) relevant to the system of interest and therefore would represent the equivalent effort of many requirements.

The resulting survey questions are reproduced in Figure 11.

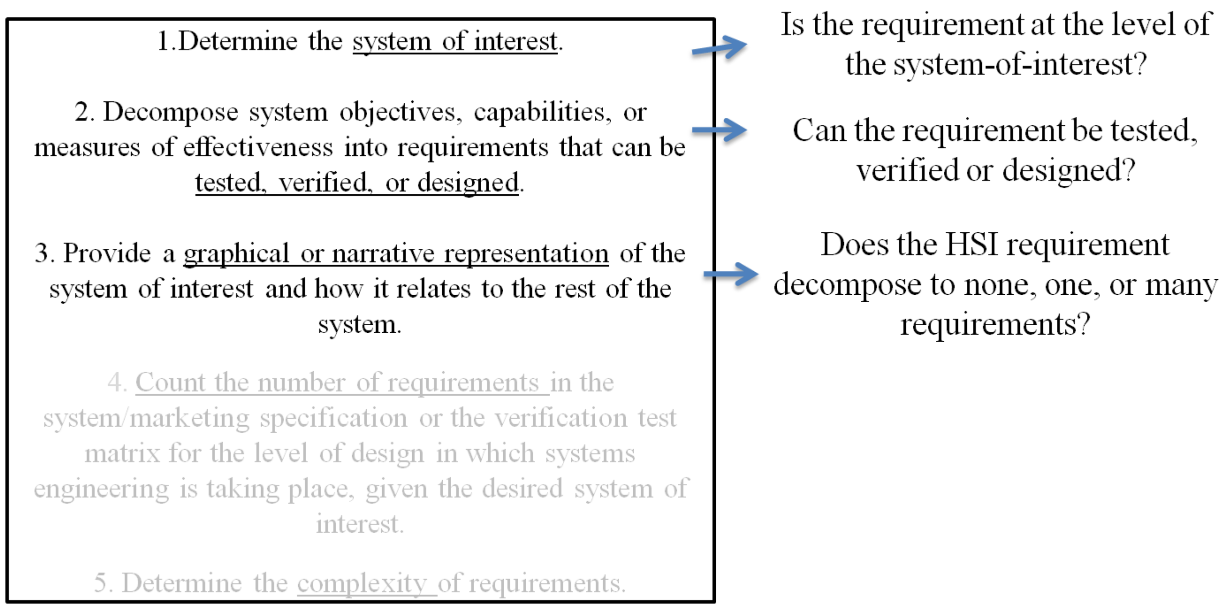


Figure 11. Development of survey questions.

4.1.3 Participant Demographics

Eight women and 8 men participated in the workshop. Participants came primarily from academia. Affiliations are shown in Table 6. Two participants omitted demographic information.

Table 6. Workshop 1 participant affiliations

Industry	5
Aerospace Corporation	2
Lockheed Martin	1
Pratt & Whitney	1
Raytheon	1
Academia	8
MIT	3
University of Bath, UK	1
University of MD	1
USC	3
Research/Consulting	1
Institute for Defense Analyses	1

The participants were asked to complete a pre-experiment survey to gather some descriptive data on the participants and their professional experience. The questionnaire can be found in Appendix 7.1. Participants were asked to rate their familiarity with Systems Engineering, Requirements Management/Engineering, Requirements Decomposition, COSYSMO, and Human Systems Integration. A Likert scale with three levels was used for the first three measures. Results are reproduced in Figure 12.

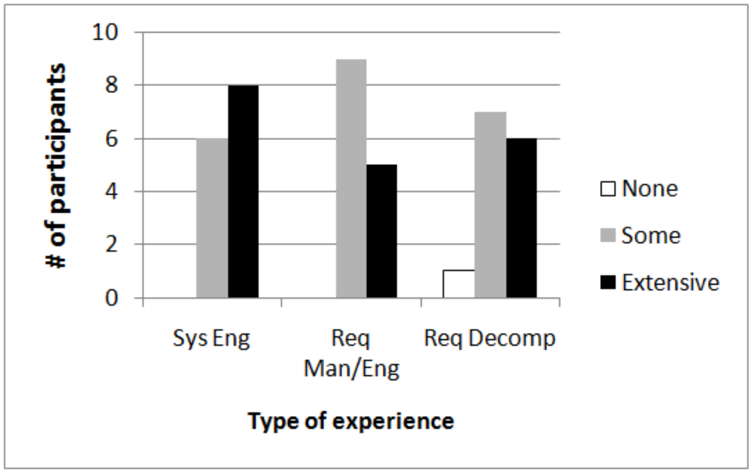


Figure 12. Workshop 1 participants' experience in systems engineering, requirements management/engineering, and requirements decomposition.

Because participants' familiarity with COSYSMO and with HSI were expected to be more varied, a five-level Likert was used in both cases. The results are reproduced in Figure 13.

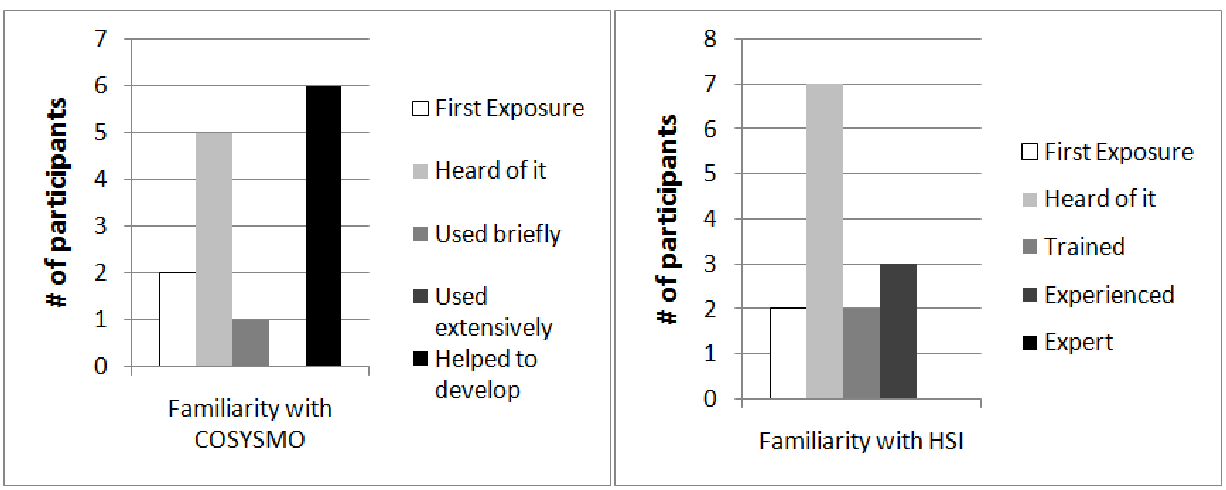


Figure 13. Workshop 1 participants' experience with COSYSMO and HSI.

None of the participants in the exercise had worked on the same programs, which means they were unlikely to have been influenced by past experience on an IPT together. Many other factors, such as gender, age, and experience, could not be controlled for.

The data show that most of the respondents thought highly of their knowledge of systems engineering and requirements. Familiarity with COSYSMO, however, was more varied. Six respondents considered themselves experts in COSYSMO or had worked on its development, but seven respondents reported little familiarity with the model.

4.1.4 Data/Results

4.1.4.1 Hypothesis #1: Producing Requirements Counts with High Reliability

The first survey question asked participants to judge "is this requirement at the level of the glass console? If not, is it too high or too low?" Instructions to the participants were adapted from Cockburn's software use case hierarchy (Cockburn 2001). The quantitative data taken from participants' responses are presented here.

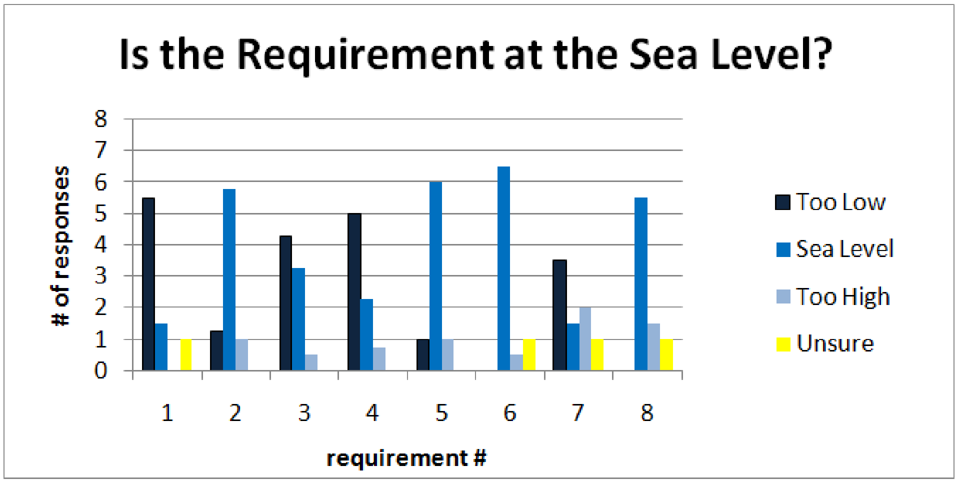


Figure 14. Survey question 1 results.

The results showed that consensus between participants largely depended on the requirement being considered. As the data show, participants tended to agree about requirements 1, 2, 5, 6, and 8. Participants agreed that the answer was one of two choices in requirements 3 and 4. Requirement 7 showed participants clearly disagreeing.

The second survey question asked participants to provide a "yes" or "no" response separately as to whether each of the requirements in question could be tested, verified, or designed. This section generated the most feedback from participants, as different stakeholders understood these terms to have different meanings. The participants' responses, both quantitative and qualitative, were binned into three categories after the workshop, as shown in Figure 15.

.

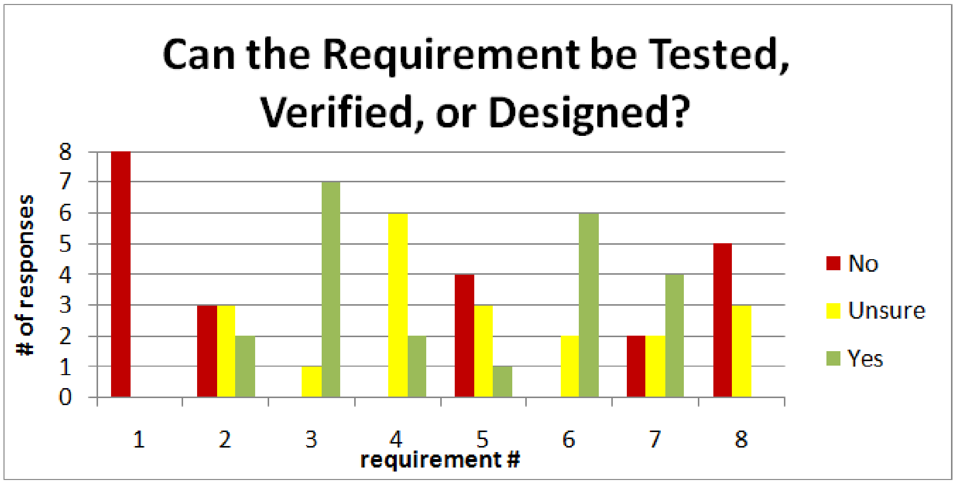


Figure 15. Survey question 2 results.

Once again, a spectrum of answers appears. Participants agreed about requirements 1,3 and 6, but were unsure about 2, 4, 5, 7, and 8.

4.1.4.2 Hypothesis #2: Helping Users Quantify HSI Requirements for Cost Estimation

The third survey question asked participants whether the requirement in question should correspond to *zero*, *one*, or *many* requirements at the level of the system of interest. Respondents were not asked to estimate an exact figure for "many requirements" as such an estimate would have required more time and analysis than was available.

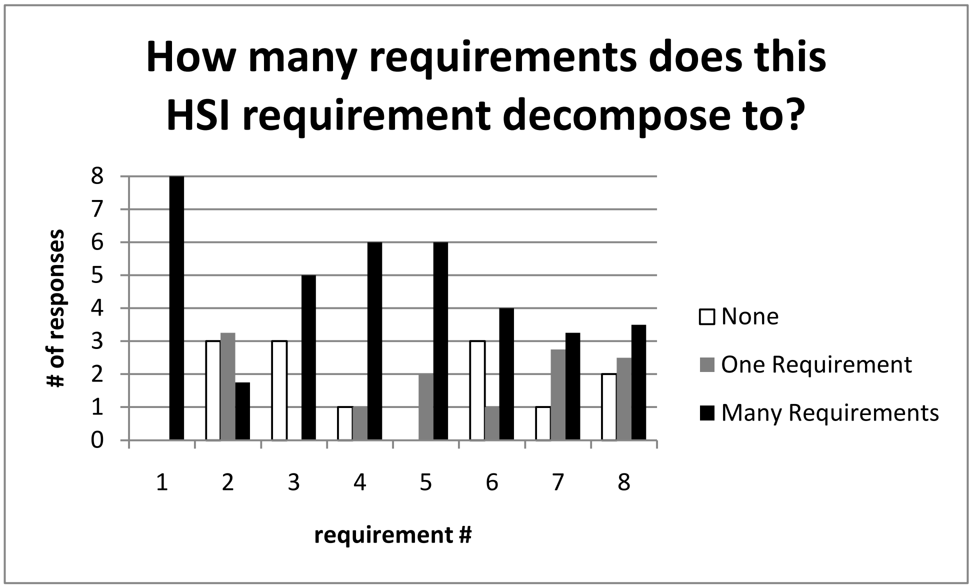


Figure 16. Survey question 3 results.

In this phase, participants agreed on requirements 1,3, 4, and 5, but disagreed about the other requirements.

4.1.5 Discussion

Hypothesis #1: Using the cost estimation decomposition steps will produce requirements counts that are common across users.

As described in the previous section, participants reached a majority consensus in 7 out 8 instances for survey question 1. Participants reached a majority consensus in 5 of 8 instances for survey question 2. However, consensus was highly dependent on the requirement in question. When the threshold for consensus is put higher, for example a 75% majority rather than 50%, consensus is reached in only 2 of 8 instances for survey question 1 and 3 of 8 instances for survey question 2.

Qualitatively, participants struggled with the distinction between "system-of-interest" and the concept of "sea level". Participants understood the two concepts separately but did not see how they related.

Participants also found it difficult to decide whether a requirement tested, verified, or designed. Participants expressed varying views on the definition of each term. Domain experience also affected answers. For example, some participants thought that a requirement that electrical systems not fail during the lifetime of a system was not testable, but others with experience in reliability engineering thought that the requirement could be verified through analysis.

Resulting proposed modifications to the COSYSMO counting rules that resulted from these insights are discussed in section 4.1.6.2.

Hypothesis #2: The cost estimation decomposition steps will help users quantify the number of HSI requirements to be input into COSYSMO.

The least agreement was seen on survey question 3. Participants came to a majority consensus on four of the eight requirements tested. Each of time consensus was reached, participants responded that the requirement should be associated with "many requirements" worth of effort. The hypothesis is therefore only partially supported. Given the broad ranges participants were given to bin their estimates in, consensus was poor. However, the counting rules did identify a need to better count HSI requirement effort.

The major point of conflict between participants was whether or not the human could be considered within system boundaries. As a result, many participants chose to rate HSI requirements as the equivalent of no requirements because they felt that the requirements would not have affected the system-of-interest. Nonetheless, in every case in which the pairs reached a majority consensus, that consensus was that the requirement in question should decompose to "many" equivalent requirements for counting purposes. While this result could be influenced by the specifics of the requirements themselves, it warrants further study.

4.1.6 Conclusions

4.1.6.1 Threats to Validity

In the course of the workshop, it became apparent that even systems engineers with high amounts of experience with requirements management and engineering could come to very different conclusions about the same high-level requirement. This was in large part due to the fact that the requirements used in the workshop were at times intentionally vague, redundant, or wordy; essentially, the exercise requirements were not ideal requirements.

These requirements were chosen because they were similar to ones that had been reviewed from government-furnished documents and were validated by experts in industry practices. Stakeholder requirements, particularly those that appear in draft form early in a system's development, cannot be expected to be ideal from a requirements engineering perspective. COSYSMO counting rules are meant to allow the early cost estimator or systems engineer to better grasp the expected need for systems engineering in a system using only whatever imperfect requirements are available.

Participants often had different perspectives on how a particular requirement should be read. They also disagreed about how to define terms like "system-of-interest", "test", and "verify". Some used the definitions they had been trained to use in industry. Others used applied natural-language definitions. Once again, however, these conditions are similar to those that might be found on a real-world requirements IPT. Stakeholders who define early requirements are not always systems engineers; in fact, they seldom are (Eigner, Haesner et al. 2002).

4.1.6.2 Modifications to COSYSMO counting rules

The observations taken from the workshop have resulted in proposed changes to the COSYSMO counting rules.

In counting rule 1, a more clear distinction between the terms "system-of-interest" and "sea level" should be made. Determining the system-of-interest establishes whether a given requirement is relevant. Determining sea level establishes whether a relevant requirement is at the correct level of decomposition to be counted in COSYSMO.

In survey question 1, participants were asked to determine whether a given requirement pertained to the system of interest. However, the framework used to quantify this measure, Cockburn's use case hierarchy, addresses whether a requirement is at the "level" of the system-of-interest. Many participants correctly commented that higher-level requirements could affect the effort associated with the system-of-interest, but not be at its "sea level". As a result, the first counting rule should be split into two, and application of Cockburn's hierarchy with respect to the system of interest should be made explicit in the second step.

In counting rule 2, the guideline should be changed to read "Assess Ability to Verify or Design" and to summarize the guidance provided on these terms by the Defense Acquisition Guidebook and other sources.

The Defense Acquisition Guidebook (2009) defines verification as the process that "confirms that the system element meets the design to or build-to specifications as defined in the functional, allocated, and product baselines." Verification can be accomplished through 4 different means: (1) Demonstration, (2) Inspection, (3) Analysis, and (4) Test. Verification principally happens during the test and evaluation phase of systems development, but systems engineers are responsible for constant verification of requirements throughout the life cycle of a system.

In certain cases, verification of a requirement within the context of a system of interest may not be possible, but the requirement may still impact systems engineering effort needed. Consider, for example, if the system of interest was a car steering wheel and the requirement being considered dealt with crash safety. The designers of the steering wheel should make design considerations for safety, but verification of safety requirements would probably be best done as part of vehicle-level crash tests.

Systems engineers using the COSYSMO counting rules need to be able to determine whether an HSI or other nonfunctional requirement is relevant to the system-of-interest. Once this is done, a derived requirement at the highest level must be written. This derived requirement must then be decomposed to a set of sea level requirements to be counted in COSYSMO. Counting 3 is therefore actually a multi-step process.

The following count rules modifications are recommended:

1. **Determine the system of interest.** For an airplane, the system of interest may be the avionics subsystem, the engine or the entire airplane depending on the perspective of the organization interested in estimating HSI. This key decision needs to be made early on to determine the scope of the COSYSMO estimate and identify the requirements that are applicable for the chosen system.

    a. **Decompose system objectives, capabilities, or measures of effectiveness down to the level of the system of interest.**

2. **Assess whether the requirements can be verified or designed.** The decomposition of requirements must be performed by the organization using COSYSMO. The level of decomposition of interest for COSYSMO is the level in which the system will be designed and tested.

3. **Provide a graphical or narrative representation of the system of interest and how it relates to the rest of the system.** This step focuses on the hierarchical relationship between the system elements. This information can help describe the size of the system and its levels of design. It serves as a sanity check for the previous two steps. In some cases, DODAF diagrams are an adequate approach (US Department of Defense 2007).

    a. Determine the impact of nonfunctional requirements on the system-of-interest

    b. Follow steps 1 and 2 for any requirements derived in step 3.a.

Since counting of requirements for COSYSMO ultimately leads to an input to a cost model, the output of the model can in turn help systems engineers justify the cost and benefit of quality requirements to decision-makers. Better requirements ultimately result in a higher-performing system.

4.1.6.3 Insights for Follow-Up Work

This workshop confirmed the need for better counting rules, particularly for HSI and other nonfunctional requirements. Participants expressed frustration and confusion over their tasks, either because the sample requirements were not written well or because they did not understand the decomposition guidelines. Such frustrations evoke the experiences of actual requirements engineers working on real systems.

Qualitatively, participants were immediately able to judge the “quality” of a requirement. In several cases, participants objected to answering a question on the grounds that the requirement presented would need to be rewritten in order for the question to be answered. These results highlight the need for the guidelines presented herein. The requirements presented in these exercises were all modeled off of actual requirements found in government documents. As such, there would likely have not been opportunity to modify the requirements. The requirements decomposition guidelines help to mitigate issues in requirements quality by quantifying the systems engineering impact each requirement poses. Understanding these impacts helps to identify those requirements that are least understood or have least quality.

Two needs were identified in the course of the workshop.

1. Better tools for consensus-building.

While participants agreed on the answers to the survey questions presented to them in many cases, there were differences in interpretation that could not be accounted for. For example, participants had different definitions of the terms “test,” “verify,” and “design”. These differences could not be captured in the quantitative data, so it is unclear if answers might have changed had differences been mitigated.

2. Further exploration of the impact of HSI on effort.

The data showed that participants were most likely to associate the requirements presented during survey question 3 with the equivalent effort of “many requirements”. It is unclear whether these results were due to the impact of human considerations or simply because the requirements happened to be difficult. Further study of the effect of HSI on effort is warranted.

4.2 Workshop 2: Estimation of Relative HSI Effort

4.2.1 Overview

Each year, the USC Center for Systems and Software Engineering (CSSE) hosts an Annual Research Review (ARR). ARR’s held during recent years have featured COSYSMO workshops, during which users and researchers share insights and work to improve the model.

The 2010 COSYSMO workshop was held March 11th, 2010. The first half of the workshop explored the integration of Human Systems Integration (HSI) into COSYSMO.

4.2.2 Research Design

4.2.2.1 Research Question/Hypothesis

The research question explored in the workshop was developed from the existing hypothesis and sub-hypotheses of this thesis:

> H1: Human Systems Integration effort can be estimated as a function of total Systems Engineering Effort.
>
> > H2: Human Systems Integration effort can be estimated by counting "number of HSI-relevant requirements"

As described in section 4.1.6.3, two needs emerged after Workshop 1: (1) better tools for consensus building, and (2) further exploration of the impact of HSI on effort.

A research question was developed in response to these needs and in following with the existing hypothesis and sub-hypothesis.

Research Question: Can an IPT reach a consensus about the effort associated with an HSI-related requirement during early cost estimation?

A testable hypothesis was then developed from the research question:

Hypothesis: Given an HSI-related requirement, an IPT can reach consensus about the amount of HSI work needed to satisfy the requirement, as compared to a nominal requirement.

4.2.2.2 Methods

Method Selection: Previous work has shown that even experienced systems engineers often disagree about how requirements should be counted for cost estimation purposes. The method applied in this workshop needed to build consensus between the participants.

Procedures: The Wideband Delphi method was applied during the workshop. The following steps were adapted taken, adapted from (Boehm 1981; Wiegers 2000).

1. Distribute data collection forms (see Appendix 7.2)
2. Introduce task and background information
    a. Purpose and goals of workshop
    b. Counting rules for cost estimation
    c. Definition of HSI
    d. Definition of a "nominal requirement"
3. Participants complete round 1 of estimation task individually
4. Facilitator plots results
5. Participants review results
    a. Participants discuss assumptions, questions, conflicts
    b. Participants modify round 1 inputs as necessary
6. Participants submit round 2 of estimation task
7. Facilitator plots results
8. Participants review results
    a. Participants discuss assumptions, questions, conflicts
    b. Participants modify round 2 inputs as necessary
9. Participants submit round 3 of estimation task
10. Facilitator plots and analyzes results

The Estimation Task: Little research has been done to address the impact of HSI on requirements. The most comprehensive resource on the subject is the Air Force HSI Office's HSI Requirements Pocket Guide, which was used in the design of the estimation task.

The example the Pocket Guide provides of a "Good HSI Requirement" was used as the nominal requirement in the estimation task. The requirement is reproduced here:

> **Example of a nominal requirement**
>
> **Threshold**: Operators shall be able to read the XX display (where XX display is the system being designed) during day and night, with no visible signature at night from 10-50m. Device must enable operators to keep head up while reading data on computer.

Participants were then asked to compare ten other requirements taken from the Pocket Guide to the nominal requirement. Requirements were sourced from Joint Requirements Oversight Council (JROC)- approved Capabilities Development Documents (CDDs) and Capabilities Production Documents (CPDs).

Participants chose from four comparative levels of effort:

1x	2x	4x	8x

Where 1x corresponded to a requirement approximately equal in effort to the nominal requirement, 2x corresponded to a requirement approximately double the effort of the nominal requirement, and so on. Available effort estimates doubled so that the change represented by moving from one estimate to an adjacent estimate was equal, regardless of start position. That is, if a participant put "1x" during Round 1 and wanted to change her answer to "2x" during Round

2, such a change would have been as if she had started at “2x” and wanted to change to “4x”. In this way, the starting point associated with each requirement should not have affected the development of consensus.

4.2.2.3 Relationship to previous work

The Wideband Delphi method has been applied in the development of COSYSMO and is the recommended method for capturing complexity ratings from stakeholders. However, assessing the complexity of a requirement is the last step in the requirements counting process. Being able to assess complexity assumes that the requirement in question has already progressed through the previous four steps effectively. Unfortunately, the data from Workshop 1 show that stakeholders faced with ambiguous high-level requirements find it difficult to come to a consensus on how a requirement should be decomposed, prior to assessing complexity.

The use of the Wideband Delphi method described in this workshop would be most useful in dealing with particularly challenging requirements that a complexity rating may not sufficiently characterize.

4.2.3 Participant Demographics

Eight men and 3 women participated in the workshop. Participants came primarily from industry. Affiliations are shown in Table 7.

Table 7. Workshop 2 participant affiliations

Industry	9
Northrop Grumman	3
Aerospace Corporation	3
Boeing	2
Raytheon	1
Academia (USC)	1
Consulting (Sofstar Systems)	1

The participants were asked to complete a pre-experiment survey to gather some descriptive data on the participants and their professional experience. The questionnaire can be found in Appendix 7.2. Participants were asked to rate their familiarity with Systems Engineering, Requirements Management/Engineering, Requirements Decomposition, COSYSMO, Human Factors/Human Factors Engineering and Human Systems Integration. A single Likert rating scale was used for each measure:

First exposure | Heard of it | Trained | Experienced | Expert

Results are reproduced in Figure 17.

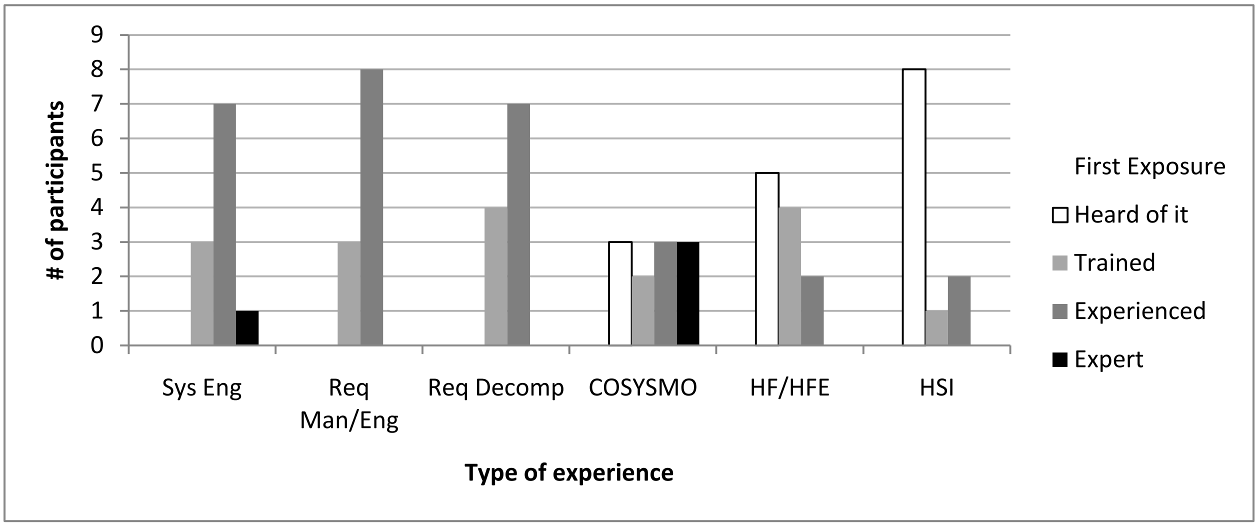


Figure 17. Workshop participants' experience in relevant fields.

The participants were all trained, experienced, or expert in the fields of Systems Engineering, Requirements Management/Engineering, and Requirements Decomposition. There was an almost even split in familiarity levels with the COSYSMO cost model, which suggests that some participants attended the COSYSMO workshop to learn more about the model, while others attended to present research or offer guidance. 6 of 11 participants were trained or experienced in Human Factors or Human Factors Engineering, but 5 of 11 had only heard of these terms. The majority of participants (8 of 11) had only heard of Human Systems Integration.

4.2.4 Data/Results

4.2.4.1 Change between Rounds

The Delphi method seeks to build consensus between participants. However, whether or not consensus emerges can depend on the task or question being estimated. The data collected from participants for each requirement are reproduced here to allow the responses to each individual requirement to be compared.

One of the 11 participants of the workshop arrived too late to participate in Round 1, so the data related to that participant has been removed. Demographic and qualitative data take the participant into account, as the participant contributed to group discussions between rounds.

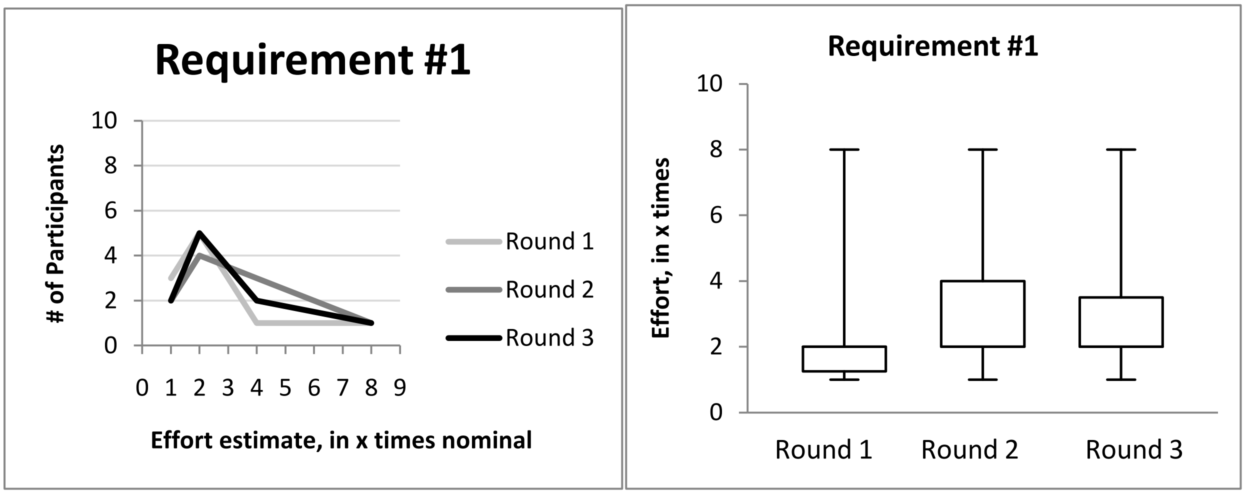


Figure 18. Workshop 2 requirement 1 results.

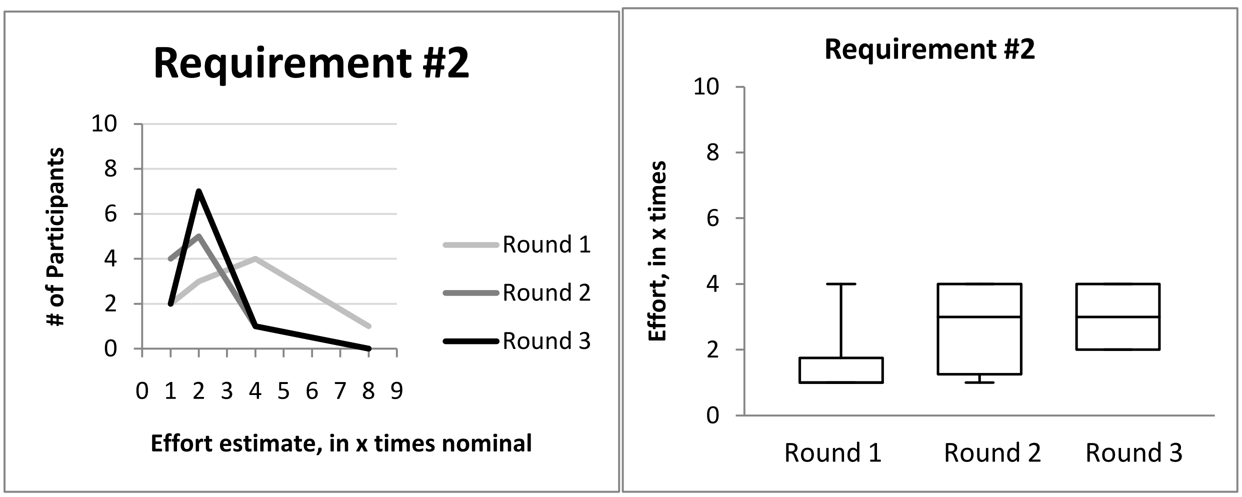


Figure 19. Workshop 2 requirement 2 results.

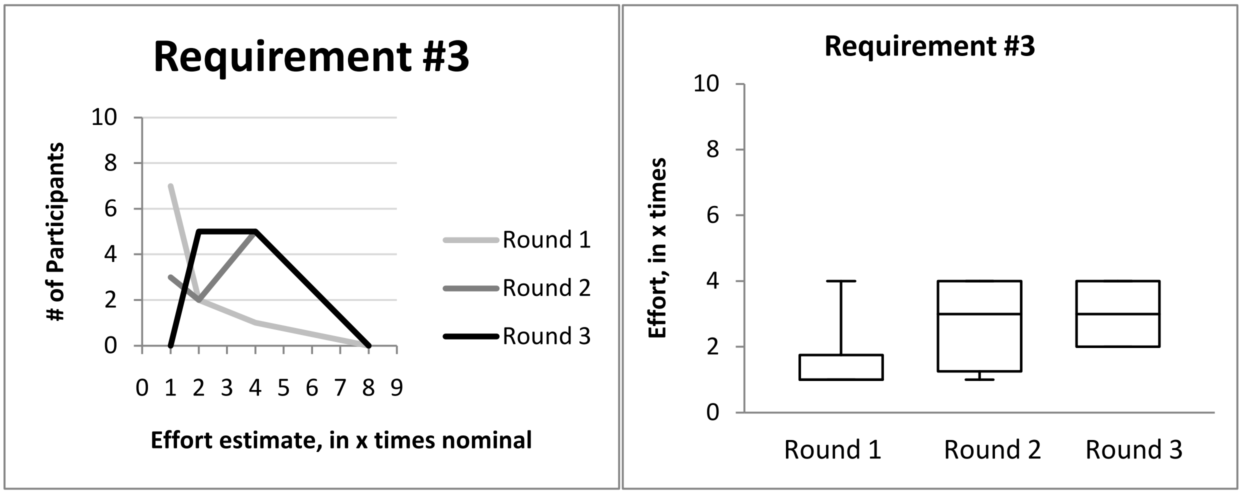


Figure 20. Workshop 2 requirement 3 results.

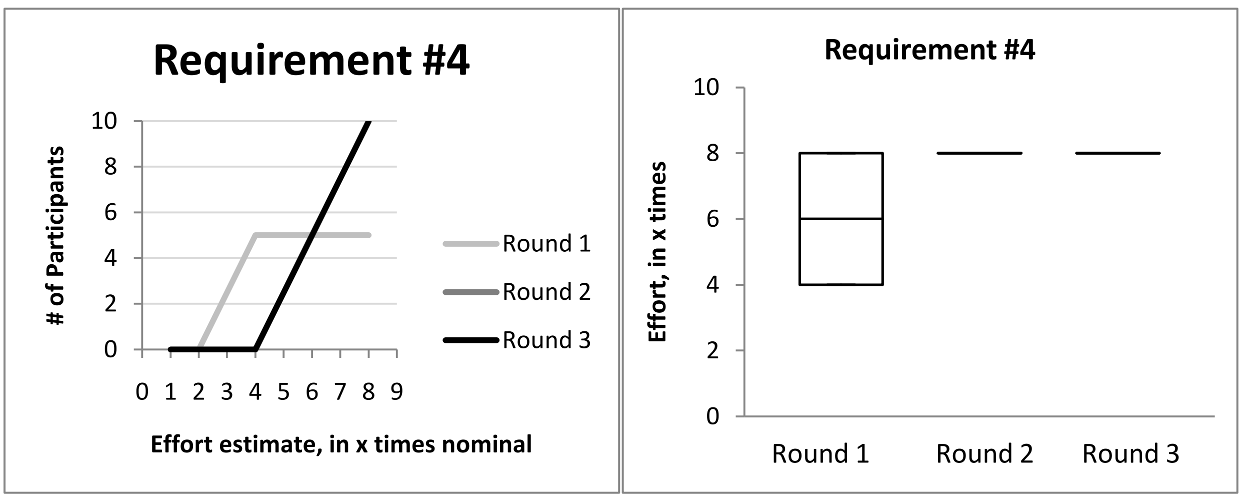


Figure 21. Workshop 2 requirement 4 results.

*The results for Requirement #4, Round 2 are obscured by the results from Round 3, indicating no change between rounds.

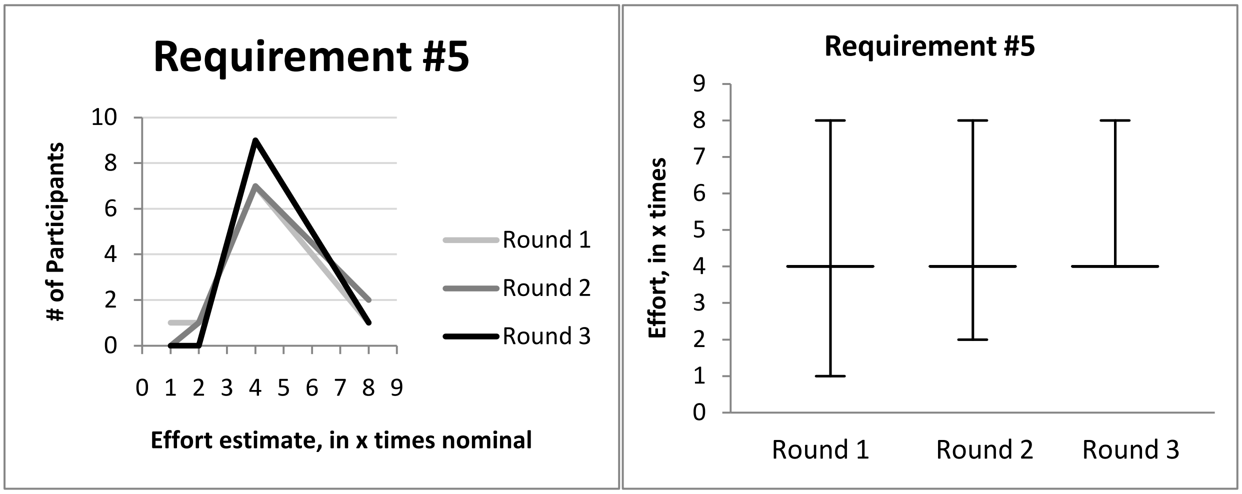


Figure 22. Workshop 2 requirement 5 results.

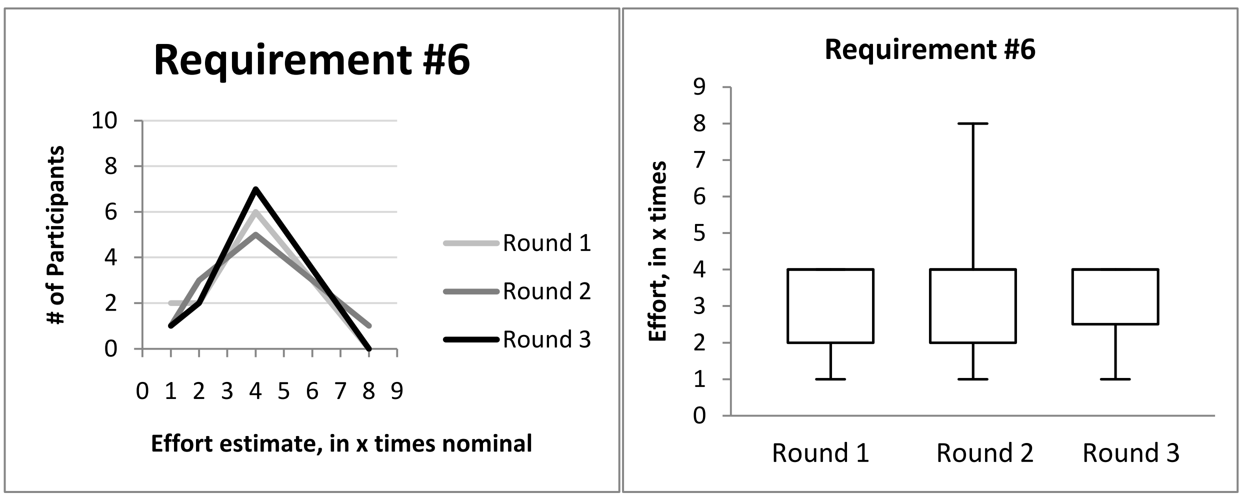


Figure 23. Workshop 2 requirement 6 results.

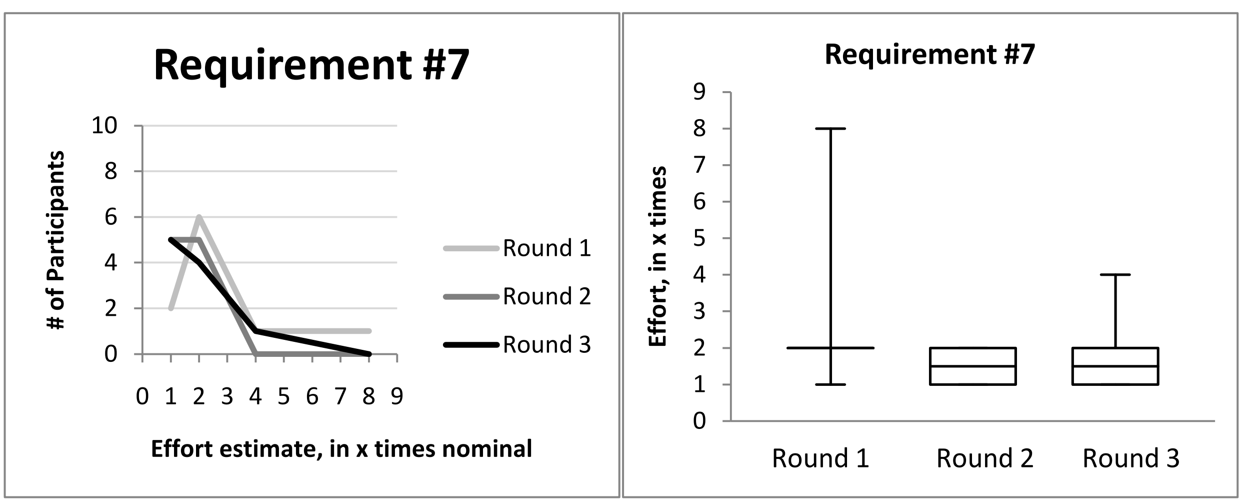


Figure 24. Workshop 2 requirement 7 results.

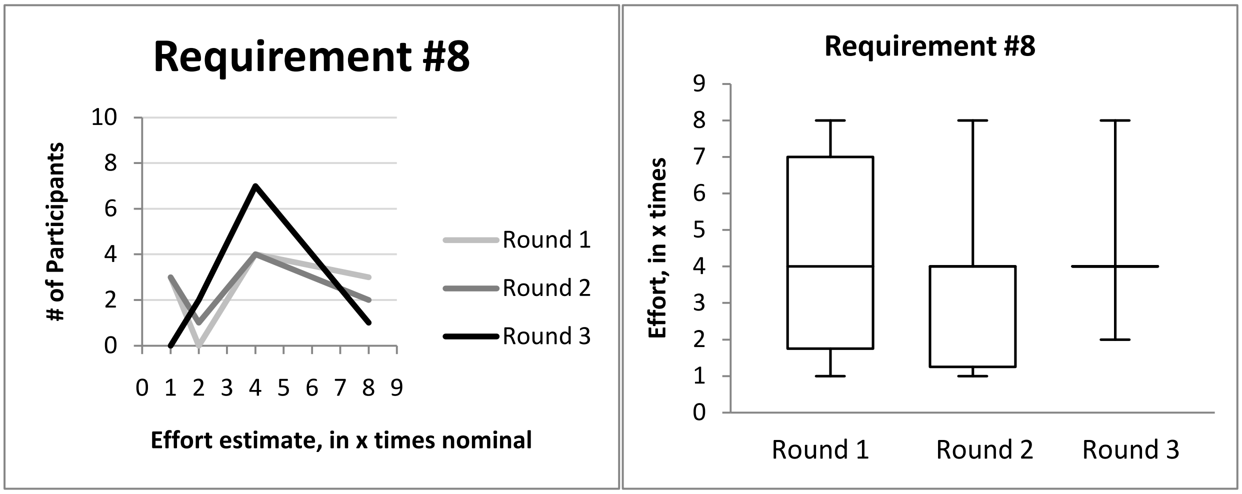


Figure 25. Workshop 2 requirement 8 results.

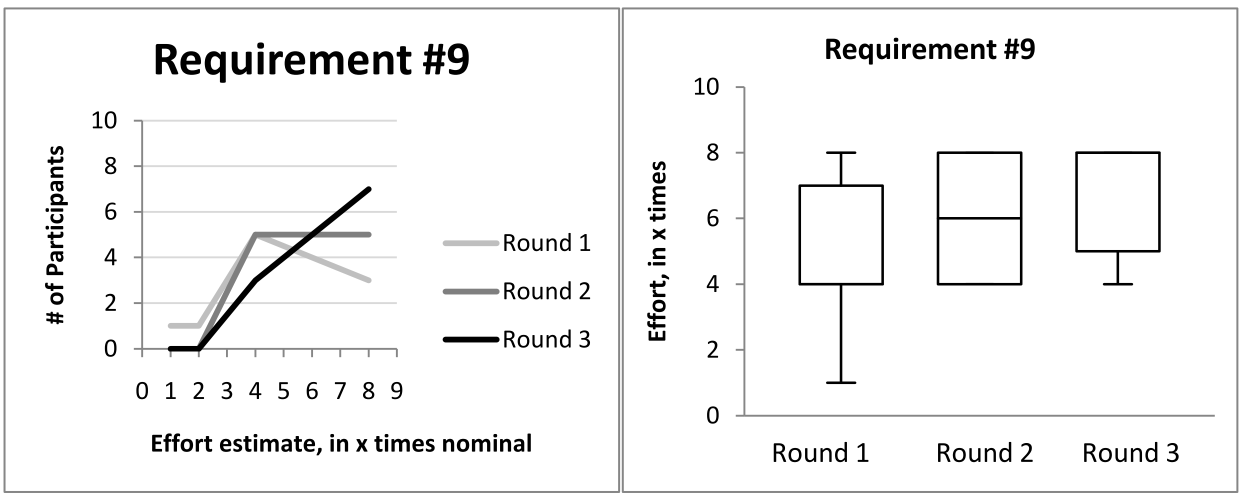


Figure 26. Workshop 2 requirement 9 results.

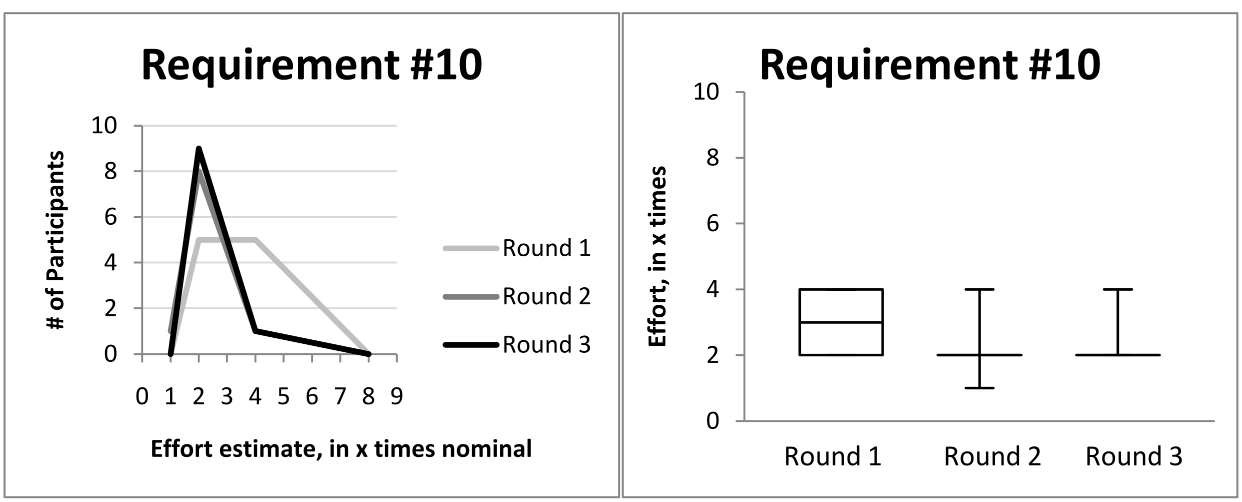


Figure 27. Workshop 2 requirement 10 results.

The left graph for each requirement shows how participants changed their effort estimates between rounds. The box plots in the right graph for each requirement shows how participants' decisions affected measures of central tendency, namely median and range.

These graphs show that in general, consensus between participants increased after each round and that the range of estimates participants offered also decreased.

4.2.4.2 Measures of Central Tendency

The previous section showed how participants' estimates were influenced after each round. To show the cumulative effect of the Wideband Delphi method across all requirements analyzed, the standard deviations of each round of analysis for each requirement was calculated and normalized. The values were normalized because requirements needing more effort can be expected to have higher absolute standard deviations. The results are reported in Figure 28.

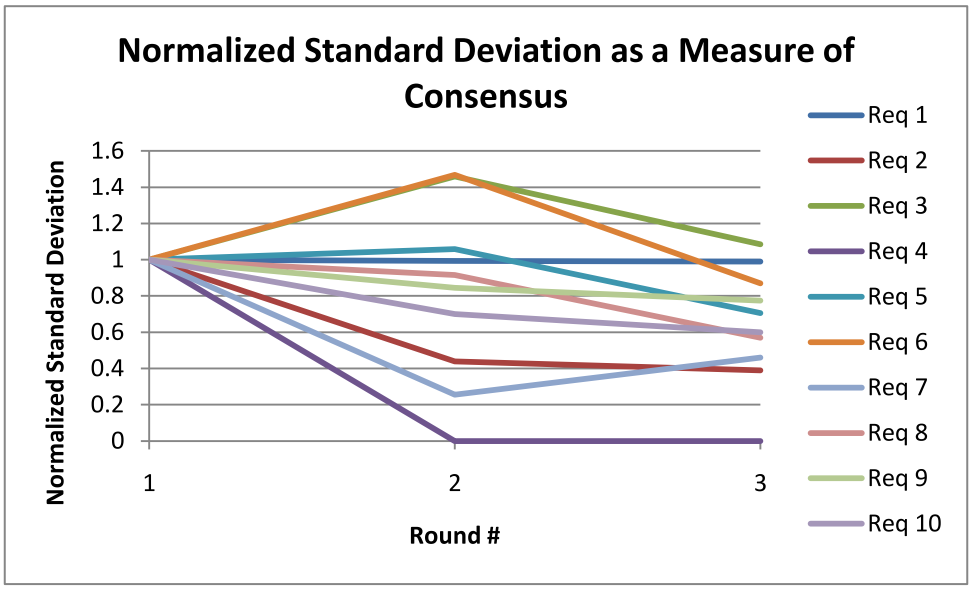


Figure 28. Normalized standard deviation as a measure of consensus.

Normalized standard deviation decreased between Round 1 and Round 2 in seven out of ten cases. It decreased in nine out of ten cases between Round 2 and Round 3. Looking at the total picture, normalized standard deviation decreased in nine out of ten cases between Round 1 and Round 3.

Standard deviation shows whether participants converged toward a particular answer; it does not show whether participants changed collective predictions. Changes in central tendency reflect collective changes; two measures are graphed in Figure 29.

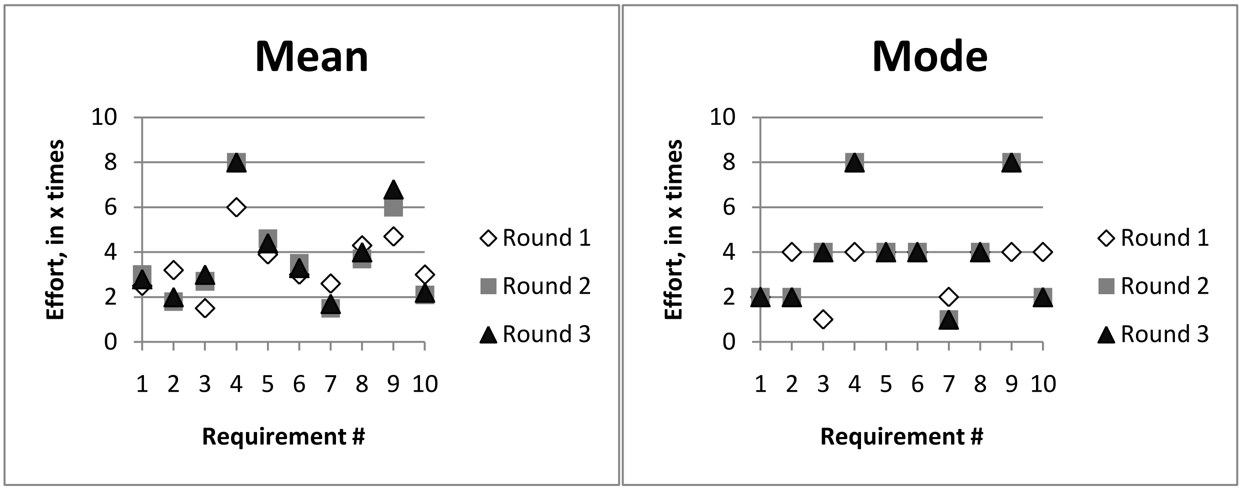


Figure 29. Mean and mode change over rounds.

The maximum mean change between Round 1 and Round 3 was a difference of 2.1x (Requirement #9, from 4.7x to 6.8x). Normalized to the mean value from Round 3, maximum change in means between Round 1 and Round 3 was 60% (Requirement #2, from 3.2x to 2x). Average change in mean across requirements was 29.4%.

Changes in mode represent a shift of all participants from one estimation value to another. Mode changed between Round 1 and Round 3 in six instances. Mode did not change between Round 2 and Round 3 in any cases.

4.2.4.3 Chapter 6 vs. Chapter 14/15 Requirements

HSI practitioners often find that as budgets and schedules compress, program managers marginalize HSI efforts in order to fund higher priority requirements. In DoD requirements documents, requirements are mainly expressed in one of three chapters. Chapter 6 of a CDD contains key performance parameters (KPPs) and key system attributes (KSAs), the two highest priority types of requirements. Chapters 14 and 15 contain lower-priority requirements. HSI considerations are often expressed in these chapters.

Systems engineers understand that requirements designated as higher-priority by the government will require more investment. However, they often make the mistake of undervaluing requirements from chapters 14 and 15 simply due to their location in a CDD.

In this workshop, five requirements were taken from chapter 6 of actual CDDs or CPDs and five requirements from chapters 14 and 15. The graphs shown in Figure 30 compare final Round 3 effort estimates between the two categories of requirements.

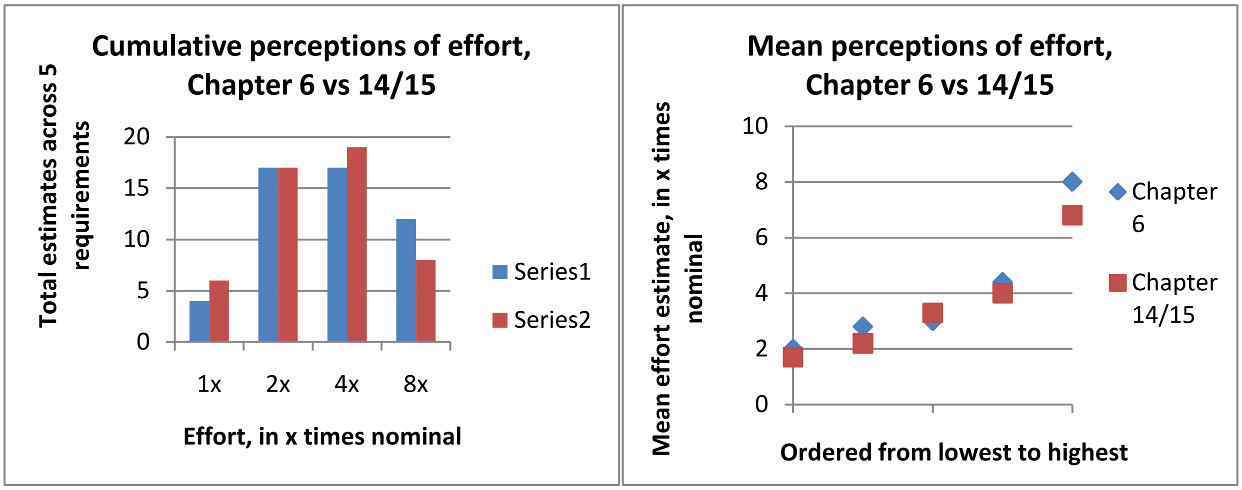


Figure 30. Perception of effort, Chapter 6 vs. Chapter 14/15 requirements.

The left graph shows that participants gave roughly the same amounts of 2x and 4x ratings to chapter 6 requirements as they did to chapter 14/15 requirements during Round 3 of estimation. Participants gave more 8x ratings to chapter 6 requirements and more 1x requirements to chapter 14/15 requirements.

The right graph compares mean effort estimates during Round 3, ordered from lowest mean to highest mean. The graph incorporates the effect of individual requirements while showing that chapter 6 requirements do not appear to be perceived as more effort-intensive across the board.

4.2.5 Discussion

4.2.5.1 Consensus-building

In this exercise, participants were given a challenging task: to estimate the effort that would be required of systems engineers to satisfy a requirement, given very little information about how the system might be built or employed. Estimates made under these conditions will unavoidably be subject to risk, uncertainty, and volatility. Using a consensus-building method such as the Wideband Delphi method encourages individuals to express and mitigate differences, addressing potential risk areas early on.

As shown previously, consensus around one answer improved in 9 out of 10 cases. However, this result does not make any claims as to whether the participants had selected the "right" answer, only that they tended to agree on a particular answer. The nature of prediction, however, is that there is no "right" answer, since the costs will not be known until long after the estimation exercise ends. Instead, the methods shown in this exercise should be applied in addition to other method. Delphi surveys can also be conducted multiple times with different groups, to assess differences in their conclusions.

The exercise was audio-taped and comments were extracted from the transcript. There were three types of discussions that helped to drive consensus.

1. Specific questions about the meaning of requirements

These questions dealt with acronyms, abbreviations, or interpretation of wording. For example, one participant asked about what a Mission-Oriented Protective Posture (MOPP) suit was. In another instance, participants discussed the differences between how the verbs "shall," "must," and "will" should be interpreted.

2. Understanding of Systems Engineering/HSI

Some participants initially recorded high estimates on some requirements because they had mentally included the cost of tests, maintenance, and design within "systems engineering." One participant stated:

> *"To me, systems engineering to me is just managing the people, organization and technology."*

This interpretation resonated with participants and participants adjusted their estimates going forward.

In terms of HSI effort, those more experienced with HSI helped to differentiate HSI effort as a subset of systems engineering from human-related work in general. One participant stated:

> *"…I mean if you think of it from an HSI perspective, then even if you ignored the difference between the "will" or the "shall," I actually think this is one area where from a systems engineering perspective, HSI is actually going to be a lot of effort. This is a lot of what they do; they bring the people in that are going to use the system and they have them try it out and they consider that and they assess it…"*

3. Differences in experience or standard practice

Participants had varying experiences with many of the domains and disciplines covered in the requirements provided for the workshop. One exchange illustrates how personal experiences were communicated and contributed to consensus:

> *Participant 1: "In my mind, making a training module is not nearly as difficult as designing an electrical piece of hardware, but it touches so many other folks and it has such a large human element in it, you really got to give yourself a lot longer schedule, and not unusually a lot more money."*

> *Participant 2: "I think I did it actually lower, you know, because usually I think of training as something that you worry about after the system's already developed or kind of set, so you're really not adding effort anywhere within the systems engineering."*

4.2.5.2 Threats to Validity

As the Delphi survey was conducted as part of an existing workshop, little control could be exercised over the composition of participants. Participant demographics are as reported, above. Likewise, because the workshop was limited by time, only three rounds of estimation could be conducted. However, additional rounds may not have helped to build consensus.

Participants may have had different mental models of the term "nominal". Participants were many times more likely to label a requirement as being equivalent to "2x" or "4x" the effort of a nominal requirement as they were to label it "1x". While the specific nature of the requirements used may have been a factor in this result, by definition nominal requirements should make up the majority of all requirements, so more ratings of "1x" should have appeared. Even if participants did overestimate, however, the ratings relative to each should be the same. That is, a requirement rated as "8x" should still be four times as effort-intensive as a requirement rated "2x", regardless of overestimation bias.

Participants stated that in some cases, 8x equivalent effort was not sufficient. Three participants wrote in their comments that requirement #4 was equivalent to more than 8x the effort of a nominal requirement. One participant wrote the same about requirement #5.

In terms of choosing the bounds for this exercise, the levels of 1x through 8x were thought to be values that participants could effectively differentiate between. For example, the 16x level was omitted because participants likely could not have effectively differentiated between it and, say a 12x requirement, which is halfway between 16x and 8x. Alternative scales could have been used and should be explored in future work.

4.2.6 Conclusions

Two significant conclusions can be drawn from the results.

Conclusion 1: Given an HSI-related requirement, an IPT can reach consensus about the amount of HSI work needed to satisfy the requirement, as compared to a nominal requirement: hypothesis supported.

Although it was clear from the qualitative feedback gathered during the workshop that participants were able to mitigate differences between rounds and therefore increase consensus, such a result is more difficult to show quantitatively. Section 4.2.4.2 showed how standard deviation of results changed between rounds. Decreasing standard deviation helps estimators identify a mean effort estimate and assess the confidence they have in that estimate. In other situations, however, a simple vote may determine an estimate, prediction, or decision.

Had a vote been applied in the workshop, a simple majority would have emerged for every requirement tested, as shown in Figure 31. During Round 1, an average of 5.4 participants chose the most popular response in their estimates. By Round 2 that number had increased to 5.8, and by Round 3 on average 7.1 participants chose the most popular answer.

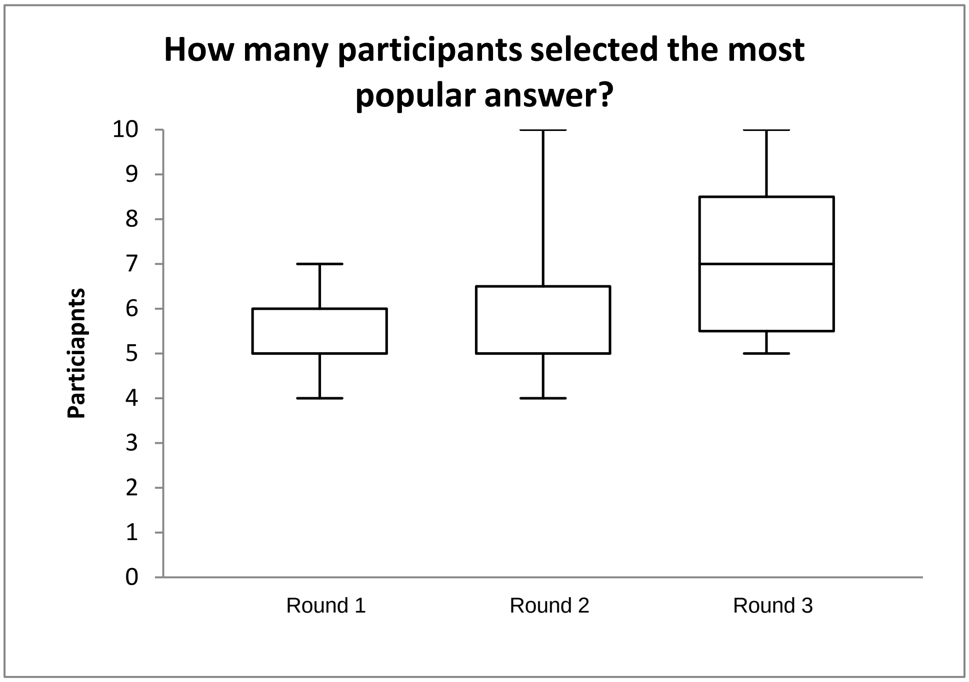


Figure 31. Consensus around one answer over time.

These results clearly show that consensus built between participants. However, the hypothesis was also concerned with the estimation of HSI work. Could the estimates that the participants agreed on be used directly as an input into COSYSMO?

The Wideband Delphi Process could be used in two ways to better account for HSI effort within COSYSMO.

1. If the process were performed to the point that consensus was clearly reached, the effort estimates could be integrated into the counting of “will’s,” “shall’s,” and “must’s”. That is, instead of counting these words for the requirements analyzed with the Wideband Delphi method, the resulting effort estimates could be substituted.
2. The Wideband Delphi process could be used to quickly identify requirements that need to be iterated. For example, if a requirement was expected to correspond to a large amount of effort because it was badly written, the requirements IPT might use that information to rewrite, decompose, and assess again.

The two ways in which the Wideband Delphi might be used rest upon the scales used in the estimation task. If, for example, participants were given ten choices, ranging from 1x to 10x, they would need to think carefully to select their answers, but those answers would be more precise than a scale with option of only 1x, 5x, and 10x.

<u>Conclusion 2:</u> No statistical difference was seen between Chapter 6 and Chapter 14/15 requirements. Implication: ignoring KPP/KSA status, industry sees HSI requirements from chapter 14/15 as just as effort-intensive as chapter 6 requirements.

Requirements from both Chapter 6 and Chapter 14/15 of CDDs/CPDs were used because these are common places where HSI-related requirements are found. The data showed that participants found the effort related to each type of requirement to be similar. There was not enough evidence to argue that the two forms of requirements were actually alike in effort. However, the evidence clearly did not support the argument that Chapter 14/15 requirements take *less* effort than do Chapter 6 requirements. Two insights can be drawn from the evidence:

1. Ignoring whether or not a requirement is designated a KPP or KSA, industry does not differentiate between requirements from the various chapters of CDDs. Since KPPs and KSAs only come from Chapter 6, unfair emphasis may be being placed on these requirements over the requirements of Chapters 14/15.
2. Future cost estimation efforts based on early requirements must include consideration of effort related to Chapter 14/15 requirements. If those requirements are incomplete or unavailable, every effort should be made to complete or improve them.

4.3 Example Application of COSYSMO for HSI

The work presented thus far in this thesis has explored basic methods for and assumptions about modeling the cost of HSI. This section demonstrates the relevance of the research provided by walking through potential uses of the COSYSMO model, with HSI taken into consideration. For brevity, only the elements needed to understand how HSI can be counted in the model are discussed; for a more in-depth discussion of COSYSMO and its function, see (Valerdi 2008).

4.3.1 Step 1: Assess scope and purpose of estimation

COSYSMO requires three types of inputs in order to produce a useful cost estimate. Before the scope of estimation can be established, the availability of these inputs must be assessed.

1. Calibration data

COSYSMO is calibrated using data from large defense contractors working mostly on aerospace programs. Follow-on research has shown that organizations that use COSYSMO with local data calibrations achieve more useful estimates. If historical data is not available either because it was not kept or because no systems similar enough to the system being estimated exist, then COSYSMO should be used as an analysis and planning tool only, not as a cost estimation model.

2. Size Drivers

Size drivers include requirements, interfaces, algorithms and operational scenarios. The cost estimator should look at what types of documents describing these drivers already exist. The level of detail to which the drivers are expressed in these documents will determine how precise the resulting cost estimate will be. If the system is expressed using multiple types of size drivers, say, requirements as well as interfaces, both may be used. However, if the documents overlap, the size driver described in greatest detail should be used in order to avoid double counting.

3. Effort multipliers

Two types of effort multipliers are applied in COSYSMO: team factors and application factors. Team factors are driven by the stakeholders involved in a system's development, whereas application factors depend on the specific system being designed. Both these types of effort multipliers are set by ratings assigned by a principle stakeholder or group of stakeholders. The availability of data for effort multipliers therefore depends on the existence of experts who are qualified to provide ratings and the reliability of those experts.

Understanding the availability and detail of the data sources above help to define the scope and usefulness of applying COSYSMO. For example, if an estimate is needed early in acquisition, usually only a draft requirements document will be available. If the prime contractor has not yet been finalized, then many of the effort multipliers will not yet be applicable. In such a case, COSYSMO might be applied as an early sensitivity analysis tool, rather than as a way to finalize budgets. If, on the other hand, COSYSMO were to be applied by contractors preparing proposals for a program, the tool could be used to compare plans and budgets at a detailed level.

4.3.2 Step 2: Collect and interpret data

It is best to collect the inputs described in the previous section consecutively: first calibration data, then size drivers, and finally cost drivers.

The majority of the work described in this thesis pertains to the collection and analysis of the size driver "number of requirements." As such, this section will describe the step-by-step process of applying the findings in this thesis to a nominal cost estimation effort.

Assume the system-of-interest is an aircraft engine, similar to the one described in the case study in 3.2. The only available source of size drivers is a draft requirements document, written at the level of the aircraft. This means that the requirements pertinent to the engine must be identified. This work corresponds to counting rules 1 and 1a. of requirements counting, as illustrated in Figure 32.

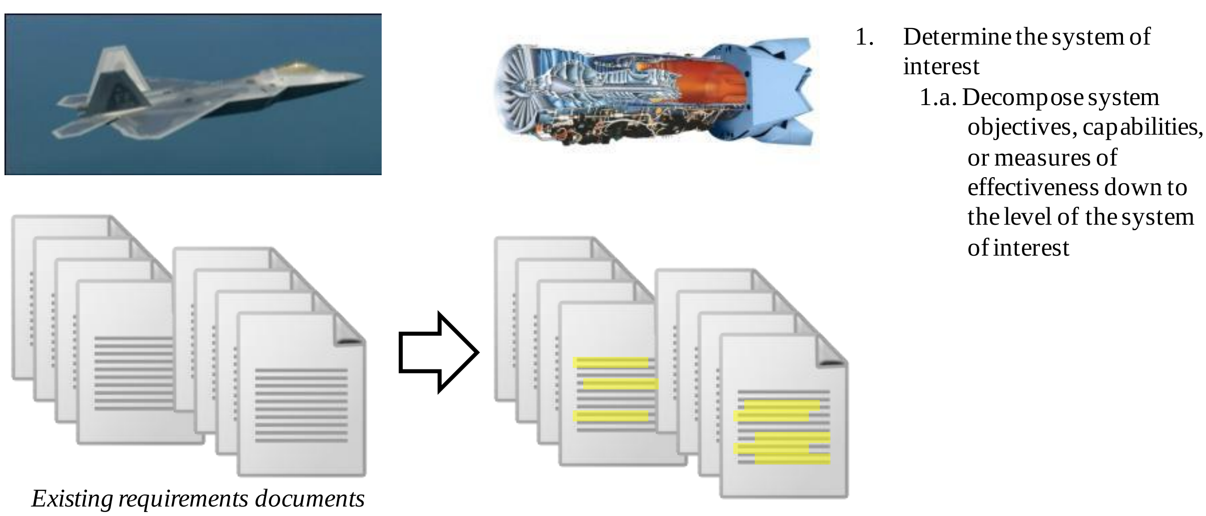


Figure 32. Counting rules 1 and 1.a.

Once the requirements relevant to the system of interest have been identified, they must be checked for quality. Requirements drive systems engineering effort, but only if there is a clear understanding of what systems engineers must do in response to a requirement.

Therefore, the next step to counting requirements asks whether the requirement being analyzed can be verified or designed. If neither is true, then there is not a clear association between the requirement and resulting systems engineering effort, so action must be taken before the requirement can be input into the model. The flowchart in Figure 33 shows several possible courses of action.

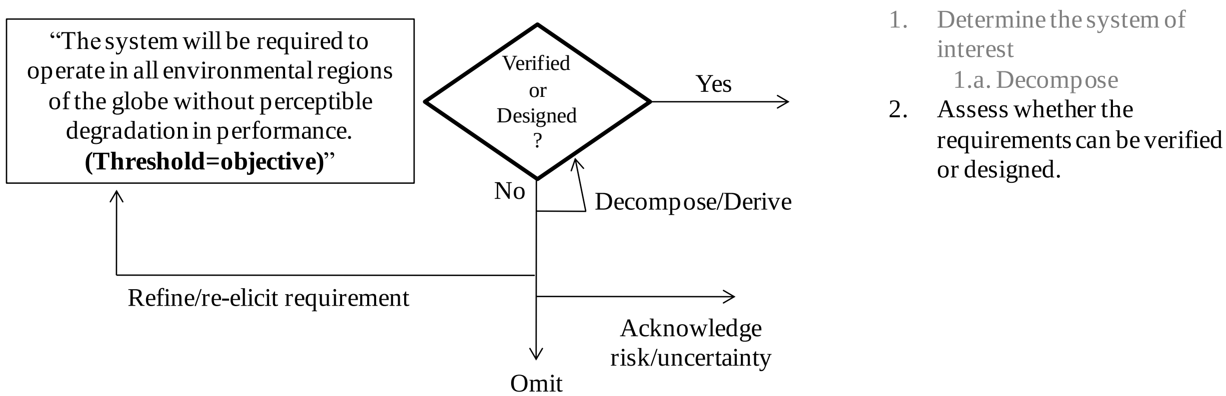


Figure 33. Counting rule 2.

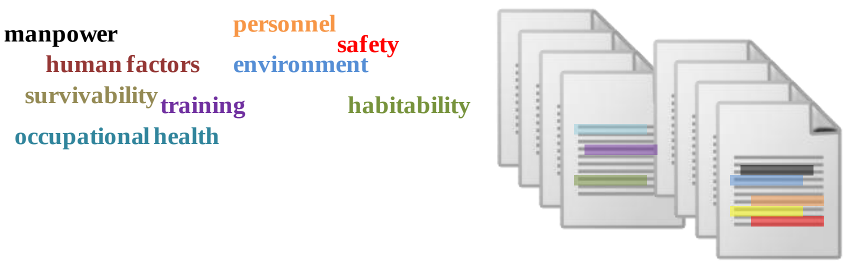
The best way to deal with a requirement that cannot be verified or designed is to engage the stakeholder and refine the requirement or elicit a new one. However, this is often not possible. The next best option is to decompose or derive the requirement into equivalent requirements relevant to the system of interest that can be verified or designed.

If the requirement cannot be derived or decomposed, then a level of risk or uncertainty should be associated with it. The work described in section 4.2.5.1 on consensus building is most relevant here. As described in section 3.3.2.4, nonfunctional requirements (like HSI requirements) are often written in ways such that it is unclear how they might be verified or designed. It was shown that the Wideband Delphi method could be used with diverse stakeholders to provide a rough estimate of risk associated with a requirement's expected effort. This method could be applied at this stage to either enter an estimated requirements count into COSYSMO, or to generate a quantitative argument for refining or re-elicit stakeholder requirements.

The last option presented in Figure 33, to omit the requirement altogether, is the worst option. This is not to say that a cost estimator might be inclined to delete the requirement from the requirements document, or even to count its effort as being equal to "0". Instead, cost estimators may be tempted to count requirements that cannot be verified or designed as one requirement. As captured in the workshops documented in both section 4.1 and section 4.2, requirements can be associated with very different levels of effort. While COSYSMO does offer the opportunity to assess the complexity of requirements later on in the requirements counting process, earlier analysis gives more fidelity to the estimate.

As discussed in section 3.3.2.4, nonfunctional requirements often result in derived requirements at the level of the system of interest. Nonfunctional requirements might be expressed at the same time as functional requirements. However, they are often imposed as constraints on a system after functional requirements or KPPs have already been defined. In any situation, nonfunctional requirements must be analyzed to determine what requirements should be derived. After completing counting rules 1 and 2, the cost estimator should analyze how nonfunctional requirements of the system can be expected to affect the requirements of the system. The example in Figure 34 shows a notional example of how HSI considerations that might be found in nonfunctional requirements could impact an existing set of requirements.

"**Human factors.** Human factors engineering principles such as specified in MIL-STD-1472 shall be employed in each XX system solution **(Threshold = Objective).**"



1. Determine the system of interest
   1.a. Decompose
2. Assess whether the requirements can be verified or designed.
3. Provide a graphical or narrative representation of the system of interest and how it relates to the rest of the system.
   3.a. Determine the impact of nonfunctional requirements on the system-of-interest
   3.b. Follow steps 1 and 2 for any requirements derived in step 3.a.

Figure 34. Counting rules 3, 3.a., and 3.b.

The analysis of nonfunctional/HSI requirements should result in a set of relevant decomposed or derive requirements. These requirements should then be analyzed using counting rules 1 and 2, as suggested by counting rule 3.b.

If the preceding steps for requirements counting have been followed correctly, then the cost estimator should be able to simply add up the resulting number of requirements and input that value into COSYSMO. Any estimates or equivalencies created as part of counting rule 2, described above, should be included in the requirement tally as necessary.

With a total set of properly counted requirements, COSYSMO provides an estimate of the systems engineering effort needed to achieve the requirements. In order to calculate the HSI effort needed on a system, a number of approaches should be considered. The diagram in Figure 35 illustrates one simple approach.

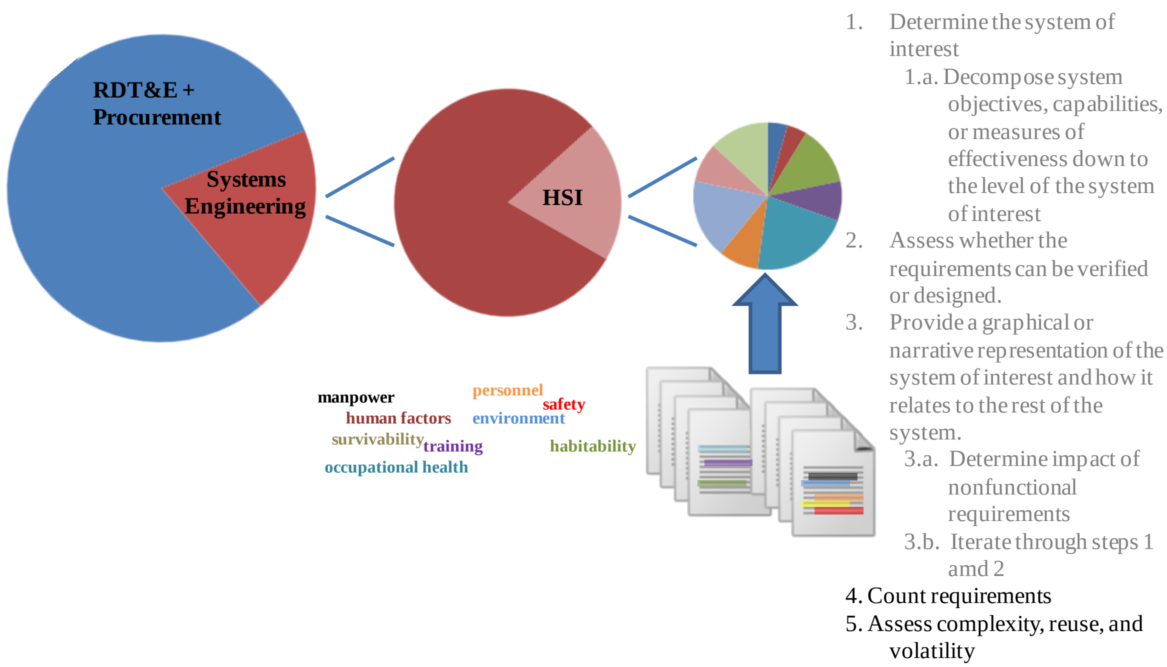


Figure 35. Counting rules 4 and 5.

A rough estimate of HSI effort can be arrived at by labeling requirements with the domain(s) of HSI they apply to. Not only can an estimate of HSI effort be made in this way, but so can estimates for the amount of effort that will be required by sub-organizations responsible for each HSI domain. Whether or not a requirement pertains to a particular HSI domain can be determined in one of two ways: (1) expert opinion and (2) keyword analysis.

Expert opinion is already used informally to identify relevant domains of HSI within IPTs. Expert opinion can also be captured using formal methods, such as with a Wideband Delphi.

Keyword analysis to identify relevant HSI domains has been successfully applied in previous research projects (711th Human Performance Wing 2008). The *HSI Requirements Pocket Guide* lists HSI keywords in chapter 8 (Air Force Human Systems Integration Office 2009b).

The methods described above do not assume that HSI should be the sole purview of any one “HSI organization” or delegated to lower-level organizations. Interviews with Pratt and Whitney engineers as well as with participants from the two workshops used in this thesis indicated that while an HSI organization or team did often exist within a company, that group was seldom the only organization responsible for HSI. Therefore, the results of a requirements analysis should not necessarily be used to fund specific organizations or an HSI team, but rather should give an indication to key decision-makers the amount of effort that will be needed to achieve key HSI results, regardless of what team carries out that work.

Complexity, reuse, and volatility are not specifically addressed in this thesis. However, a distinction should be emphasized here between the application of the Wideband Delphi method to assess complexity and the method in which it was applied for Workshop 2.

Although complexity ultimately corresponds to increased effort in COSYSMO, it would be wrong for an estimator to think of complexity weightings in terms of "effort added." For example, if a requirement seems difficult, the decision to assign it a difficulty rating should not be based on the question "will realizing this requirement cost me more effort?" but rather on the question "does this requirement match the definition of a 'difficult' requirement?" The definitions of different complexity levels are discussed in section 3.3.1.2.

If complexity ratings were to be assigned based purely on a cost estimator's perception of effort, two problems would arise: (1) the cost estimator would be able to introduce too much bias into the estimate, and (2) there would be no clear way to track why one requirement was labeled "difficult" and another "nominal."

4.3.3 Step 3: Consider team and application factors

Once a final list of requirements has been reached, effort multipliers need to be assessed in order to address factors unique to an organization or program. The definition and assignment of effort multipliers is covered in (Valerdi 2008). Unique issues that arise with effort multipliers related to HSI and requirements are addressed here.

Consider the application of COSYSMO to a notional system. The system is defined by 200 easy, 200 nominal, and 100 difficult requirements, as shown in Figure 36. Assuming all effort multipliers are assessed to be nominal, the program will require about 300 person-months of systems engineering effort to complete.

Now consider that a high maintainability requirement was designated a KPP by the primary stakeholder. Although the impact of this KPP on decomposed requirements has already been captured, KPPs can increase awareness, tracking, and focus across on an issue across a program. Therefore, it is decided that a high rating should be assigned to the "level of service requirements" effort multiplier. Next, it is noted that sophisticated modeling tools are needed to perform HSI-related safety and occupational health analyses. However, a review of potential tools reveals a lack of suitable candidates. The decision is therefore made to decrease the value of the "tool support" effort multiplier, but only in relation to safety- and occupational health- related requirements. Screenshots from the academic version of COSYSMO illustrating this process are shown in Figure 36 and Figure 37. Academic COSYSMO is available at cosysmo.mit.edu.

CO SYS MO 2.0 — CONSTRUCTIVE SYSTEMS ENGINEERING COST MODEL

4-Nov-09

ENTER SIZE PARAMETERS FOR SYSTEM OF INTEREST

Reuse

	Easy	Nominal	Difficult	equivalent size
# of System Requirements	200	200	100	800.0
# of System Interfaces				0.0
# of Algorithms				0.0
# of Operational Scenarios				0.0
				800.0

SELECT COST PARAMETERS FOR SYSTEM OF INTEREST

Requirements Understanding	N	1.00
Architecture Understanding	N	1.00
Level of Service Requirements	N	1.00
Migration Complexity	N	1.00
Technology Risk	N	1.00
Documentation	N	1.00
# and diversity of installations/platforms	N	1.00
# of recursive levels in the design	N	1.00
Stakeholder team cohesion	N	1.00
Personnel/team capability	N	1.00
Personnel experience/continuity	N	1.00
Process capability	N	1.00
Multisite coordination	N	1.00
Tool support	N	1.00
composite effort multiplier		**1.00**

SYSTEMS ENGINEERING PERSON MONTHS: 303.0

Figure 36. COSYSMO example original estimate (Fortune 2009).

If all effort multipliers are left at nominal, the corresponding effort is calculated to be 303 person-months.

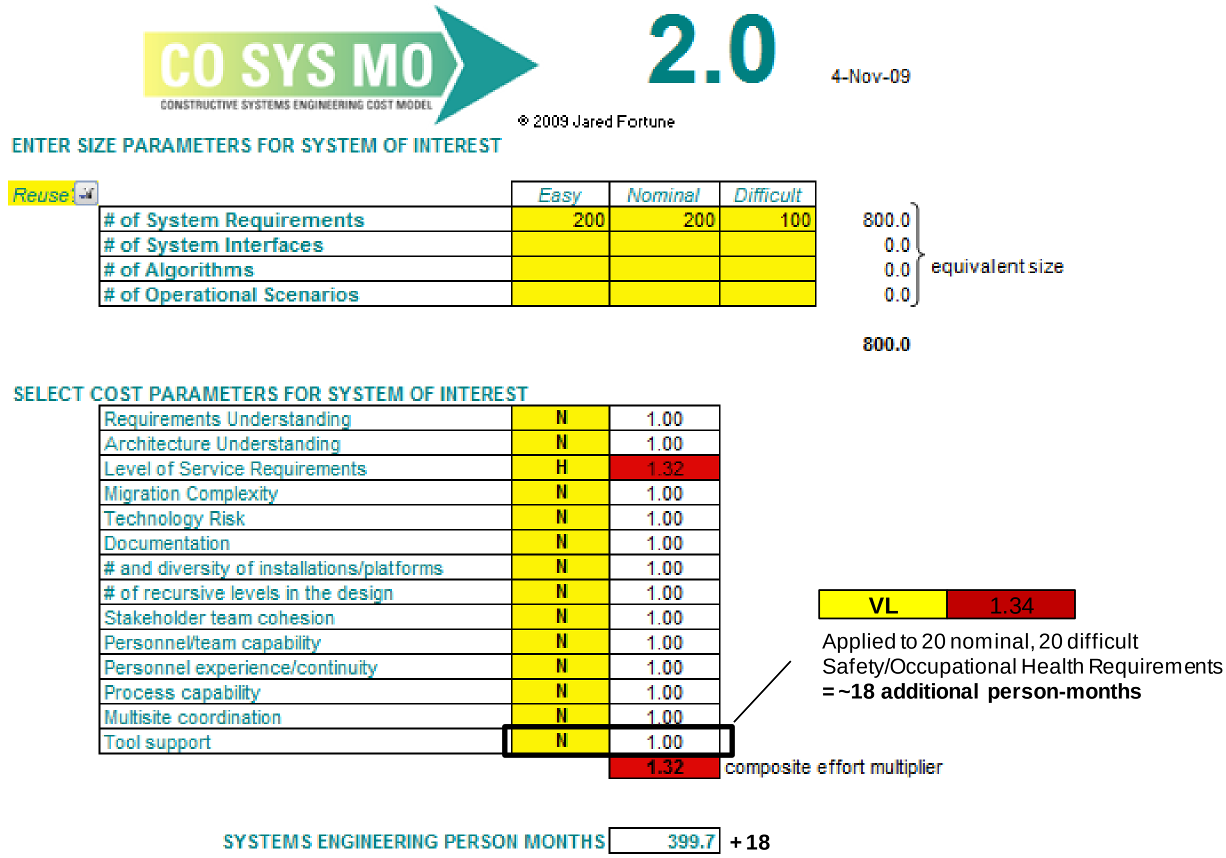


Figure 37. Estimate taking into account HSI considerations (Fortune 2009).

Figure 37 shows how the level of service requirements and tools support can affect an effort estimate. By increasing level of service requirements to high, the total effort estimate increases to 399.7 person-months from 303.0. This effort multiplier is assessed over the program's entire body of requirements, but the tools support effort multiplier should only be applied to safety and occupational health requirements. Assume that there are 20 nominal and 20 difficult safety/occupational health requirements. In order to calculate the impact of very low tools support on just these requirements, 20 nominal and 20 difficult requirements are input into the model. The level of service requirements is set to high. The tools support effort multiplier is then varied between nominal and very low, and the impact on effort is compared. Applying this process results in an increased effort of 18 person-months.

The processes described in this section can be tailored to fit a particular organization's business practices. For example, in one company safety might be its own engineering group while in another it might be split into sub-groups. Effort multipliers can be made to apply only to certain sets of requirements, as necessary. It is important to keep in mind, however, that as the focus of estimation changes, calibration data must also be adjusted.

4.3.4 Step 4: Evaluate estimate and iterate

The power of cost estimation often comes not from the number generated by a model, but the process through which the estimate is reached. The production of a cost estimate forces estimators to seek out and validate sources of data. The results of cost estimation may highlight program risks or and ambiguity. Cost estimates should not be considered a tool for producing a static budget, but rather a vehicle for communication. Cost estimates must be performed iteratively, for variables can and do change.

This chapter highlighted some of the key ways in which HSI can affect systems engineering cost and shows how a parametric cost estimation tool like COSYSMO can translate those impacts into relative differences in cost. It also draws attention to the importance of understanding systems engineering requirements and their decomposition. The next chapter offers some concluding thoughts and provides guidelines for future development.

5 DISCUSSION AND RECOMMENDATIONS

5.1 Summary of Methods and Results

Figure 38 summarizes the methodology followed in this thesis, showing how initial research questions and hypotheses motivated the work done.

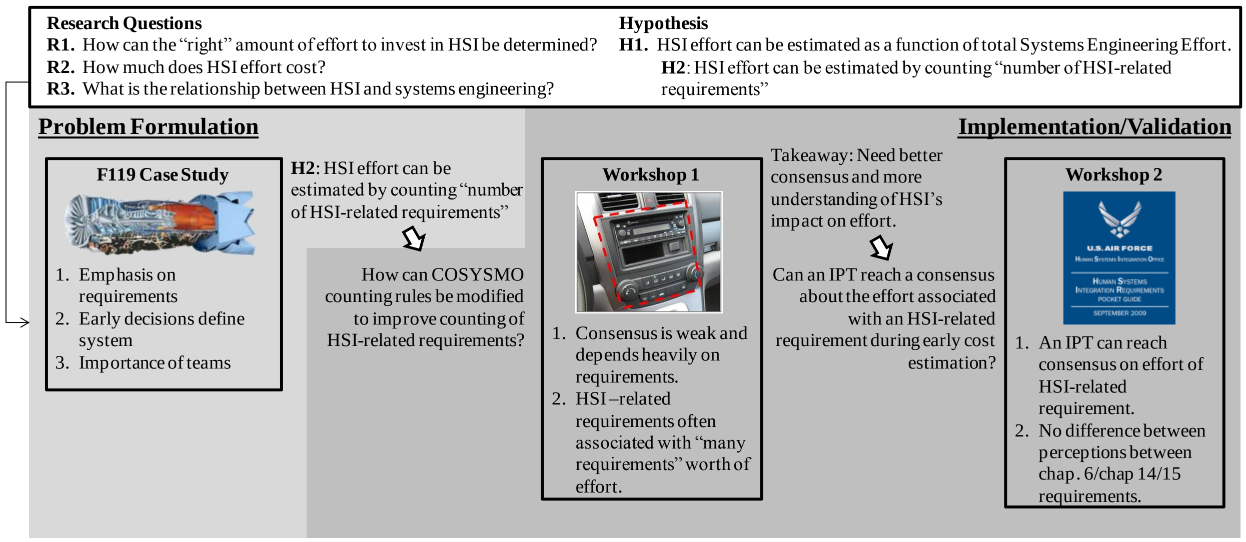


Figure 38. Thesis structure.

As summarized in section 1.2.1, the research was initiated by a set of research questions and an initial hypothesis, represented as R1, R2, R3, and H1. These research questions and hypothesis informed the problem formulation phase, during which a sub-hypothesis, H2, was generated. H2 emphasizes the importance of requirements counting for effort estimation. It informed the inputs introduced into existing requirements counting rules, described in section 4.1.6.2.

The implementation and validation phase described the testing of the proposed changes to requirements counting. The research question for Workshop 1 was informed by the conclusions of the case study. The conclusions drawn from Workshop 1 then informed the research question developed for Workshop 2. The conclusions from both workshops informed an example application of COSYSMO for HSI, described in section 4.3.

At each point in this research, work done was motivated by the overarching hypotheses and research questions. Figure 39 illustrates how conclusions generated during each phase contributed to these initial objectives.

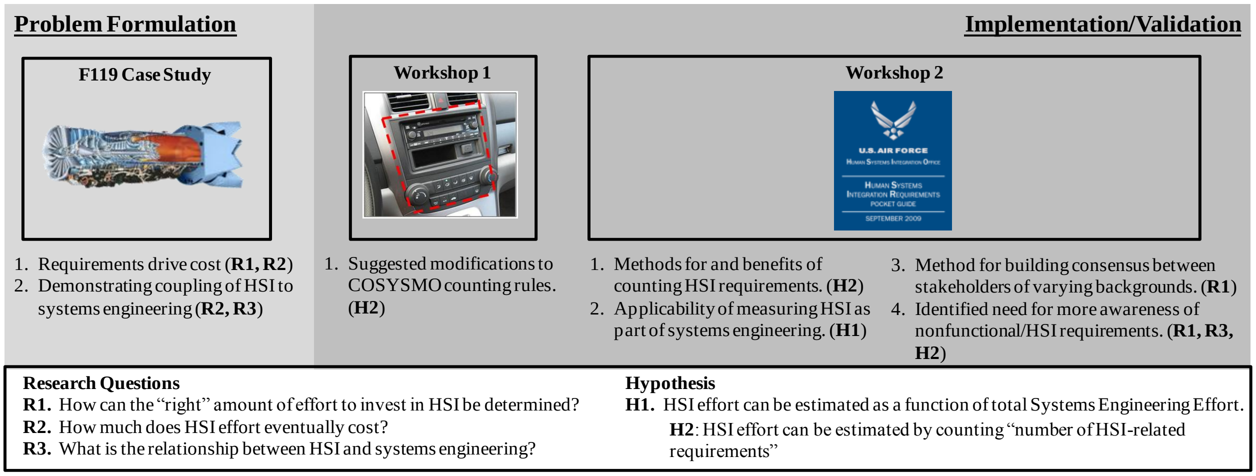


Figure 39. Linking conclusions to research questions and hypotheses.

The following sections describe how work done for this research has supported each of the initial research questions and hypotheses.

5.1.1 R1: How can the "right" amount of effort to invest in HSI be determined?

How can a program manager guarantee HSI considerations will be adequately addressed in a program, without putting an unnecessary burden on budget? In this research, the case study of the F119 showed that requirements drive contractor effort, both in terms of specific deliverables and in terms of high-level focus. In addition, it showed that requirements must be established early on to make a lasting impact. The two workshops highlighted the different ways in which stakeholders interpret requirements. Workshop 2 in particular emphasized the need to better account for the impact of nonfunctional and HSI-related requirements. Therefore, the question of how best to determine the "right" amount of HSI effort deals not with a dollar amount, but with engineering the "right" requirements early on in development. For example, cost models like COSYSMO will be able to provide an accurate estimate of HSI effort needed, but only if that estimate is based on the "right" inputs.

5.1.2 R2: How much does HSI effort cost?

The F119 case study showed that HSI effort could not be isolated from systems engineering effort after the fact. The example application of COSYSMO for HSI in section 4.3 shows how HSI effort can be estimated. The question of how much HSI costs depends largely on how HSI is organized within an organization. What engineering group carries out HSI-related requirements? How is systems engineering carried out in the organization? Requirements engineering and COSYSMO can be used in conjunction to estimate the effort related to HSI in the context of systems engineering as a whole. How that effort is carried out and how budgets are assigned depends on the specific organization using the model.

5.1.3 R3: What is the relationship between HSI and systems engineering?

The F119 case study showed one example of how HSI was tied into systems engineering planning and practice. Workshop 2 showed that HSI considerations can exist in a variety of types of requirements that span systems engineering. Review of literature showed that many sources of systems engineering guidance already include mention of the human as a critical part of the system. From its creation, HSI was meant to be a part of systems engineering. Successful HSI depends greatly on successful systems engineering, and the inverse is also true.

5.1.4 Hypotheses: HSI as a function of systems engineering requirements

The workshops conducted as part of the implementation and validation work done for this thesis challenged participants to apply a cost estimation methodology to the estimation of HSI effort. The insights gained from those workshops were integrated into the cost model. Workshop 1 established a need to better account for nonfunctional requirements. Workshop 2 showed that HSI-related requirements and nonfunctional requirements could be counted effectively using COSYSMO counting rules. Section 4.3 gave an example of how the modified counting rules could be used to accommodate even difficult or hard-to-understand HSI requirements. As discussed above, HSI effort should not be considered as a single process of systems engineering or as a discipline separate from systems engineering. Rather, HSI touches all aspects of systems engineering, and any effort estimate for HSI must take into account how systems engineering is carried out on the program being estimated.

The work done in this thesis therefore supports both the fact that HSI effort can be estimated by counting requirements related to HSI and that HSI effort can be measured as a function of systems engineering effort.

5.2 Limitations and Future Work

The one case study described in this thesis explores work that was completed in 1991. While it proved valuable for formulating the research performed in the second half of the thesis, it does not represent a broad sampling of how HSI is currently performed in industry. Gerst and Rhodes (2010) have documented the perceptions of industry and government on HSI in ongoing work that seeks to integrate HSI into systems engineering leading indicators. Elm et al. (2008) surveyed major government contractors and subcontractors to a better understanding of systems engineering return on investment. Performing similar research specifically investigating HSI would be difficult due to varied definitions of HSI and HSI domains. Future work should investigate HSI practices at a larger sample of organizations in order to generate best practices and so better inform future cost estimation research.

The two workshops conducted during implementation and validation were both limited by the number and diversity of available participants. The benefit of gathering data from the workshops was that attendees tended to have advanced experience in systems engineering and cost estimation. However, there was no way to randomly sample. As such, no statistical conclusions could be drawn about the broader population. General conclusions could be arrived at if the insights generated from the workshops conducted in this research were used to develop survey questions that could be distributed to a larger sample of HSI and systems engineering practitioners.

Workshop 2 in particular applied the Wideband Delphi method and showed that consensus was generated between rounds. However, it was unclear whether the estimates participants agreed on would have been the same estimates a different group of participants would have agreed on.

Performing the workshop on different groups of participants would increase the strength of the of the estimates arrived at.

5.3 Impact on Policy

The objective of this research was to explore research questions related to the cost of HSI. However, estimating the investment needed upfront to enable HSI practice tells only one part of the story. Understanding HSI effort needed early on enables decisions that allow for programs to finish on budget and schedule, while delivering performance to users. This section proposes three ways in which policy can be modified to support these goals.

5.3.1 Mandate Early Systems Engineering Cost Estimate

Section 2.1.1 gives an overview of the Defense Acquisition System. An AoA is one of the first documents that must be completed at the outset of acquisition. AoA's are performed during the MSA phase, prior to milestone A. They compare the costs and operational effectiveness of different means of achieving the desired capability.

Section 3.3 of the Defense Acquisition Guide provides the most detailed guidance on how to prepare an AoA (Defense Acquisition University 2010a). Cost estimates created for the AoA must take into account total life cycle costs. The DAG suggests several ways to estimate costs. The goal of the cost analysis in an AoA is to look broadly at the cost of a system across development, procurement, and operations & sustainment.

Systems engineering budget has historically been associated with between 10 and 20% of a system's development budget (RAND Corporation 2006). Other work has shown that RDT&E makes up roughly 2% of the total cost of a system, while procurement makes up roughly 34% (Louden 2000). Given these figures, the natural conclusion would be that systems engineering budget makes up a very small fraction of the total budget of most systems. Since AoA cost analyses are performed very early on in acquisition, systems engineering costs can be difficult to estimate with precision. Early cost estimators tend to focus their efforts on what is in their minds the largest drivers of cost – hardware, software, and manpower. Systems engineering, is however, planned early on. A Systems Engineering Plan (SEP) is due at Milestone A and is continually revised through Milestone B. The existence of this document contributes to the impression that a cost estimate for systems engineering is not needed in the AoA.

The assumption that hardware, software, and manpower are the major drivers of a program's cost is wrong for two reasons. First, the bulk of systems engineering costs are associated with RDT&E and procurement budgets, which have been shown to make up roughly a third of total ownership costs. However, these numbers are based on completed, successful projects. Many projects in defense are never completed, often due to failures in management of budget and schedule, directly related to systems engineering.

The rule of thumb for RDT&E spending on a project is 2% of total life cycle costs. However, projected fiscal year 2010 RDT&E costs made up 12% of the DoD's acquisition budget (taking into account RDT&E, Procurement, O&M and personnel costs) (Department of Defense 2009). Add to this fact that systems engineering decisions made upfront drive costs later acquisition (as argued by the General Accounting Office (2003), and in the case study described in section 3.2), and it becomes clear that systems engineering should not be excluded from cost estimates conducted during the AoA.

Mandating a systems engineering cost estimate during the AoA will accomplish three main objectives:

1. Force early analysis of systems engineering and HSI issues
2. Create better link between AoA, SEP, HSIP, and Cost Analysis Requirements Description (CARD)
3. Prevent cutting budget from systems engineering/HSI activities

Although this recommendation argues that cost estimation must be done for systems engineering, it also directly addresses the HSI. Since HSI is tightly coupled with SE, the benefits of conducting early cost analysis for systems engineering will by necessity also consider HSI issues, as outlined in section 4.3.

5.3.2 Better Integrate HSI into SE Training and Guidance

Systems engineering as discipline started because challenges faced in Defense Acquisition required traditional engineers to consider how different fields of engineering contributed to the performance of an overarching system. The first systems engineers were trained engineers from traditional fields who applied their engineering skills to systems challenges (Ferris 2007). In many ways, the development of HSI has closely followed the systems engineering model. While human-related considerations have long played a part in systems acquisition, the complexity of modern systems requires a systems-thinking approach.

Similar to how mechanical engineers, electrical engineers, and software engineers became systems engineers by necessity as systems engineering demands increased, so have trained Human Factors engineers, safety engineers, and the like begun to put on an HSI "hat" as program managers have asked for this expertise. Likewise, as attention has grown on HSI, organizations have been stood up to advocate for and manage HSI within programs. While having a central advocate for HSI promotes awareness throughout each service and ensure HSI issues are addressed on major programs, no one organization is capable of conducting the reviews necessary to ensure thorough HSI practice throughout all programs.

The approaches toward HSI dictated by policy do not correlate to current execution. DoD policy states that the objective of HSI is to "optimize total system performance [and] minimize total ownership costs" while systems engineering must "balance system performance, cost, schedule, and risk" (Department of Defense 2008a). Likewise, the Defense Acquisition Guide states that "the program manager should integrate system requirements for the HSI domains with each other, and also with the total system" and goes on to emphasize the inclusion of HSI on IPTs within integrated product and process development (IPPD) (Defense Acquisition University 2010a). If, however, HSI "practitioners" or "experts" are likely, as described above, to be rooted in one of the domains of HSI, it seems unlikely a single HSI advocate on an IPT could fully address the myriad issues that arise when integrating humans into system requirements.

From the perspective of the systems engineer, HSI training is insufficient. HSI-related requirements continue to be discussed in the same breath as nonfunctional requirements and "other DOTMLPF" considerations (CJCS 2009). Basic training for requirements managers within DoD adopt this guidance as well; the chapter on systems engineering makes one mention of HSI and other mentions of HSI again consider as part of nonfunctional requirements.

The result of current policy on HSI and systems engineering is that despite the fact that HSI essentially exists to integrate human considerations into systems engineering, stovepipes are being formed where "HSI practitioners" are not necessarily systems engineers and systems engineers may not have the skills necessary to fully consider HSI impacts.

The work reported in this thesis has illustrated how important though difficult it is to analyze requirements early in development. In order to adequately consider HSI issues early, HSI must become a mandatory part of basic systems engineering training. Systems engineers are responsible for system integration; it follows that they, not a solely HSI-specific practitioner, should be responsible for the integration of human considerations into systems. This is not to say that a training module on HSI should be developed and distributed. Such an approach would be too easily ignored. Without context, the domains of HSI are complex and can be abstract. Instead, it is recommended that HSI should be written into existing systems engineering guidance. The Air Force HSI Office has developed a detailed document explaining where HSI can be inserted into systems engineering throughout life cycle processes (Air Force Human Systems Integration Office 2009a). It is recommended that similar HSI considerations should be integrated into guidelines for the SEP and the Systems Engineering Management Plan (SEMP). Currently, guidance for preparation of these documents makes no mention of HSI (Office of the Deputy Under Secretary of Defense for Acquisition and Technology Systems and Software Engineering/Enterprise Development 2008).

These recommendations to enhance integration of HSI into systems engineering guidance and training do not negate the need for HSI experts. Just as some systems will need dedicated Human Factors engineers, safety engineers, environmental engineers, and the like, so will some systems need to have HSI experts on staff. Existing efforts to ensure development of an HSI Plan (HSIP) early in acquisition should continue. The argument, therefore, is not that existing efforts are wrong, only that more can be done in systems engineering practice to ensure successful HSI.

5.4 Conclusion

This research sought to explore the cost of HSI effort within acquisition programs. The goal of the research was to integrate HSI into a systems engineering cost model in order to give decision makers an accurate picture of expected HSI effort. A case study and two workshops led to significant contributions to an existing systems engineering cost model that will enable better capture of HSI considerations.

The path of this research has also highlighted many other insights relevant to the practice of HSI. HSI has been championed in a number of high-level policy documents. Many case studies have shown how relatively small investments in HSI have resulted in significant savings. However, HSI continues to be considered a challenge to implement.

Understanding HSI costs is not important only the generation of a cost model. In a culture where program managers are solely responsible for the cost and schedule of their systems, the value of HSI must be expressed in terms of cost in order to garner adequate attention. As HSI is tightly coupled to systems engineering, early estimates for systems engineering cost must be made, so that these costs are understood and planned for early by program managers. Program managers must understand the level of investment they need to make in HSI, the benefits it will provide, and the activities that must be done to enable successful HSI.

Part of what this research has done is to show that in order to understand HSI costs, one must first understand the impact of HSI in requirements. Requirements drive system development, and therefore the majority of development costs. As requirements are a fundamental part of systems engineering practice, systems engineers must be better trained to recognize HSI issues early in the requirements engineering process. Better integration of HSI into systems engineering will help program managers understand the work that must be completed to achieve HSI. Only then will systems engineers, working together with HSI and HSI domain experts, be able to define affordable systems that deliver value to warfighters.

6 REFERENCES

6.1 Works Cited

711th Human Performance Wing (2008). Human systems integration (HSI) into joint capabilities integration and development syste: Analysis and development of HSI domain specific capabilities language and metrics. Depart of the Air Force Human Performance Optimization Division Human Performance Integration Directorate. Brooks City-Base, TX, 711th Human Performance Wing.

Air Force Cost Analysis Agency (2008). Air Force cost analysis handbook. Department of the Air Force. Arlington, VA, Air Force Cost Analysis Agency.

Air Force Human Systems Integration Office (2009a). Human systems integration in acquisition: Integrating human concerns into life cycle systems engineering. Department of the Air Force. Falls Church, VA, Air Force Human Systems Integration Office**:** 37.

Air Force Human Systems Integration Office (2009b). Human systems integration requirements pocket guide. Washington, D.C.

ANSI/EIA (1999). ANSI/EIA-632-1988 Processes for engineering a system. New York, NY, American National Standards Institute.

Aronstein, D. C., M. J. Hirschberg, et al. (1998). Advanced tactical fighter to F-22 raptor : origins of the 21st century air dominance fighter. Reston, VA, American Institute of Aeronautics and Astronautics.

Bahill, A. and S. Henderson (2005). "Requirements development, verification, and validation exhibited in famous failures." Systems Engineering **8**(1): 1-14.

Boehm, B. (1981). Software engineering economics. Upper Saddle River, NJ, Prentice Hall.

Boehm, B., C. Abts, et al. (2000). Cost estimation with COCOMO II. Upper Saddle River, NJ, Prentice Hall.

Bolkcom, C. (2007). CRS Report for Congress: F-22A Raptor. Washington, DC, Congressional Research Service.

Booher, H. R., Ed. (1990). MANPRINT: An approach to systems integration. New York, NY, Van Nostrand Reinhold.

Booher, H. R. (1997). Human factors integration: Cost and performance benefits on Army systems. Aberdeen Proving Ground, MD, Army Research Laboratory.

Booher, H. R. (2003). Handbook of human systems integration. Hoboken, NJ, Wiley-Interscience.

Bruseberg, A. (2009). Cost-benefit analysis for human factors integration: A practical guide.

Chief of Naval Operations (2009). Navy personnel human systems integration (NAVPRINT). N151. Washington, DC, Government Printing Office.

CJCS (2004). The national military strategy of the United States of America: A strategy for today; A vision for tomorrow. Chairman Joint Chiefs of Staff. Washington, DC, Government Printing Office.

CJCS (2009). CJCSI 3170.01G - Joint Capabilities Integration and Development System. J. C. o. S. Chairman. Washington, DC, Government Printing Office.

Cockburn, A. (2001). Writing effective use cases. Boston, Addison-Wesley.

Dalkey, N. C. (1969). The Delphi method: An experimental study of group opinion. Santa Monica, CA, RAND Corporation.

Defense Acquisition University (2009). Integrated defense acquisition, technolog, and logistics life cycle management system, Version 5.3.4. Department of Defense. Washington, DC, Government Printing Office.

Defense Acquisition University. (2010a). "Defense Acquisition Guidebook." Retrieved April 28, 2010, 2010, from https://dag.dau.mil/.

Defense Acquisition University. (2010b). "Defense acquisition portal: Acquisition process."   Retrieved April 28, 2010, from https://dap.dau.mil/aphome/Pages/Default.aspx.
Department of Defense (1969). Military standard: System engineering management. Department of Defense. Washington, DC, Government Printing Office.
Department of Defense (1974). Military standard: Engineering management. Department of Defense. Washington, DC, Government Printing Office.
Department of Defense (1989). Military standard: Human engineering design criteria for military systems, equipment, and facilities. Department of Defense. Washington DC, Government Printing Office.
Department of Defense (1992). DoD 5000.4-M - Cost analysis guidance and procedures. Department of Defense. Washington, DC, Government Printing Office.
Department of Defense (1994). Military standard: Systems Engineering. Department of Defense. Washington, DC, Government Printing Office.
Department of Defense (2003). DoDD 5000.01 - The defense acquisition system. Department of Defense. Washington, DC, Government Printing Office.
Department of Defense (2008a). DoDI 5000.02 - Operation of the Defense Acquisition System. Department of Defense. Washington, DC, Government Printing Office.
Department of Defense (2008b). National Defense Strategy. Department of Defense. Washington, DC, Government Printing Office.
Department of Defense (2009). National defense budget estimates for FY 2010. Department of Defense. Washington, DC, Government Printing Office.
Department of Defense. (2010, 04/18/2010). "Defense Acquisition Portal."   Retrieved 04/18/2010, 2010, from https://dap.dau.mil/.
Department of the Air Force (2006). Air Force Instruction 10-601-Capabilities-based requirements development. Department of the Air Force. Washington, DC, Government Printing Office.
Department of the Air Force. (2009). "F-22 Raptor."   Retrieved February 10, 2010, from http://www.af.mil/information/factsheets/factsheet.asp?id=199.
Department of the Air Force. (2010). "Human Performance Integration Directorate."   Retrieved April 20, 2010, from http://www.wpafb.af.mil/afrl/711hpw/hpi.asp.
Deskin, W. J. and J. J. Yankel (2002). Development of the F-22 propulsion system. 38th AIAA/ASME/SAE/ASEE joint propulsion conference and exhibit. Indianapolis, IN.
Eigner, M., D. Haesner, et al. (2002). Requirements management and traceability.
Elm, J. P., D. R. Goldenson, et al. (2008). A Survey of Systems Engineering Effectivenesss - Initial Results. NDIA SE Effectiveness Committee, Carnegie Mellon University & National Defense Industrial Association.
España, S., N. Condori-Fernandez, et al. (2009). Evaluating the completeness and granularity of funcation requirements specifications: A controlled experiment. 17th IEEE International Requirements Engineering Conference. Atlanta, GA.
Ferris, T. (2007). Some early history of systems engineering - 1950's in IRE publications (Part 1): The problem. Proceedings of the 17th Annual International Symposium of the International Council on Systems Engineering, San Diego, CA, INCOSE.
Fortune, J. (2009). Estimating systems engineering reuse with the constructive systems engineering cost model (COSYSMO 2.0). Industrial and Systems Engineering. Los Angeles, University of Southern California. **Doctor of Philosophy:** 92.
General Accounting Office (2003). Best practices: Setting requirements differently could reduce weapon systems' total ownership costs. General Accounting Office. Washington, DC, Government Printing Office.
Gerst, K. and D. H. Rhodes (2010). "Strengthening systems engineering leading indicators for human systems integration considerations – insights from the practitioner community. 8th Annual Conference on Systems Engineering Research. Hoboken, NJ.
Gille, W. H. (1988). Design to cost handbook. Department of the Army. St. Louis, MO, US Army Troop Support Command, Directorate for Resource Management, Cost Analysis Division.

Gillette, F. C., Jr. (1994). "Engine design for mechanics." SAE International.
Glinz, M. (2007). On non-functional requirements. 15th IEEE International Requirements Engineering Conference. New Delhi, India. **7:** 21-26.
Goode, H. H. (1957). System engineering; an introduction to the design of large-scale systems. New York,, McGraw-Hill Book Co.
Government Accountabiility Office (2009). GAO cost estimating and assessment guide: Best practices for developing and managing capital program costs. Government Accountability Office. Washington, DC, Government Printing Office.
Gunston, B., Ed. (2007). Jane's aero-engines. Alexandria, VA, Jane's Information Group.
Harrison, J. A. and M. J. Forster (2003). Human systems integration requirements in system acquisition. Handbook of human systems integration. H. R. Booher. Hoboken, NJ, John Wiley and Sons.
Hasselrot, A. and B. Montgomerie (2005). An overview of propulsion systems for flying vehicles. Stockholm, Sweden, Swedish Defense Research Agency.
Hewitt, G. (2003). Human factors assessments in investment analysis: Definition and process summary for cost, risk, and benefit, Federal Aviation Administration.
Höfer, A. and W. F. Tichy (2007). Status of empirical research in software engineering. Empirical software engineering issues: Critical assessment and future directions. V. R. Basili, D. Rombach, K. Schneideret al, Springer.
Hofmann, H. F. and F. Lehner (2001). "Requirements engineering as a success factor in software projects." Software, IEEE **18**(4): 58-66.
Honour, E. C. (2002). Toward an understanding of the value of systems engineering. 1st Annual Conference on Systems Integration. Stevens Institute of Technology, Hoboken, NJ.
Hull, E., K. Jackson, et al. (2005). Requirements engineering. London, Springer.
International Council on Systems Engineering (2006). The INCOSE Systems Engineering Handbook. Seattle, WA, International Council on Systems Engineering.
International Ergonomics Association. (2010, March 31, 2010). "What is ergonomics." Retrieved April 28, 2010, 2010, from http://www.iea.cc/browse.php?contID=what_is_ergonomics.
International Organization for Standardization (2005). Systems engineering — Application and management of the systems engineering process. New York, NY, IEEE Computer Society.
International Society of Parametric Analysts and Society of Cost Estimating and Analysis (2004). Parametric estimating handbook (4th edition). Vienna, VA, ISPA/SCEA Joint Office.
ISO/IEC (2002). ISO/IEC 15288:2002(E) Systems engineering - system life cycle processes. Geneva, Switzerland, International Organization for Standardization.
Kamata, M. I. and T. Tamai (2007). How Does Requirements Quality Relate to Project Success or Failure? Requirements Engineering Conference, 2007. RE '07. 15th IEEE International.
Keith, D. R., J. G. Williams, et al. (1983). Joint agreement on increased R&D for readiness and support. Washington, DC.
Klimoski, R. J. (1978). "Simulation Methodologies in Experimental Research on Negotiations by Representatives." The Journal of Conflict Resolution **22**(1): 61-77.
Landsburg, A. C., L. Avery, et al. (2008). "The art of successfully applying human systems integration." Naval Engineers Journal **120**(1): 77-107.
Laplante, P. A. (2009). Requirements engineering for software and systems. Boca Raton, CRC Press.
Louden, J. (2000). Total ownership cost: Naval sea systems command, Naval Sea Systems Command,.
Malone, T. B. and F. Carson (2003). "HSI top down requirements analysis." Naval Engineers Journal **115**(2): 37-48.
McGrath, J. E. (1966). A social and psychological approach to the study of negotiation. Studies on Behavior in Organizations: A Research Symposium. R. Bowers. Athens, University of Georgia Press.
McGrath, J. E. (1981). "Dilemmatics: The study of research choices and dilemmas." The American Behavioral Scientist **25**(2).

Ministry of Defence (2006). Cost arguments and evidence for human factors integration. Ministry of Defence, Systems Engineering & Assessment Ltd. on behalf of the Human Factors Integration Defence Technology Centre.
Müller, G. (2005). Industry and academia: Why practitioners and researchers are disconnected. INCOSE Symposium 2005. Rochester, NY.
National Aeronautics and Space Administration (2008). 2008 NASA cost estimating handbook. Washington, DC, NASA Headquarters, Cost Analysis Division.
Naval Sea Systems Command. (2009). "Improving system performance and affordability by improving sailor performance." from http://www.navsea.navy.mil/nswc/dahlgren/ET/HSI/default.aspx.
Neill, C. J. and P. A. Laplante (2003). "Requirements Engineering: The State of the Practice." Software, IEEE **20**(6): 40-46.
Nemeth, C. P. (2004). Human factors methods for design: Making systems human-centered. Boca Raton, FL, CRC Press.
Office of the Deputy Under Secretary of Defense for Acquisition and Technology Systems and Software Engineering/Enterprise Development (2008). Systems engineering plan preparation guide, Version 2.01. Washington, DC, ODUSD(A&T)SSE/ED.
Pew, R. W. and A. S. Mavor, Eds. (2007). Human-system integration in the system development process: A new look. Washington, DC, The National Academies Press.
Pratt and Whitney. (2002). "F119 Cutaway." from http://www.pw.utc.com/StaticFiles/Pratt%20&%20Whitney/News/Press%20Releases/Assets/Images/f119_low2.jpg.
Proctor, R. W. and T. Van Zandt (2008). Human factors in simple and complex systems. Boca Raton, FL, CRC Press.
RAND Corporation (2006). Systems engineering and program management: Trends and costs for aircraft and guided weapons programs. Santa Monica, CA, RAND Corporation.
Rechtin, E. (1991). Systems architecting: Creating & building complex systems. Upper Saddle River, NJ, Prentice Hall.
redjar. (2007). "center console." flickr.com, from http://www.flickr.com/photos/87434398@N00/386725173/.
Reynolds, J. C. (1983). JAFE field visit. Department of the Air Force. Wright-Patterson AFB, OH, Author.
Rhodes, D. and D. Hastings (2004). The case for evolving systems engineering as a field within engineering systems. MIT Engineering Systems Symposium. Cambridge, MA.
Roper, M. (2006). "Capabilities-based costing: Approaches to pre-Milestone-A cost estimating." MSIAC Modeling & Simulation Journal Online **4**.
Runkel, P. J. and J. E. McGrath (1972). Research on human behavior; a systematic guide to method. New York,, Holt.
Sager, L. and R. Grier (2005). Identifying and measuring the value of human factors to an acquisition project. Human Systems Integration Symposium. Arlington, VA.
US Army Cost and Economomic Analysis Center (2002). Department of the Army cost analysis manual. Department of the Army. Washington, DC, Government Printing Office.
US Army MANPRINT Directorate. (2007). "MANPRINT program posters." from http://www.manprint.army.mil/manprint/promo.asp.
Valerdi, R. (2008). The constructive systems engineering cost model (COSYSMO): Quantifying the costs of systems engineering effort in complex systems. Saarbrücken, Germany, La Vergne, Tenn., VDM Verlag Dr. Müller/produced in USA by Lightning Source.
Valerdi, R. and H. L. Davidz (2009). "Empirical research in systems engineering: challenges and opportunities of a new frontier." Systems Engineering **12**(2): 169-181.
Valerdi, R. and K. K. Liu (2010). Parametric cost estimation of human systems integration. The economics of human systems integration: Valuation of investments in people's training and

education, safety and health, and work productivity. W. B. Rouse. Hoboken, NJ, John Wiley and Sons.
Valerdi, R., K. K. Liu, et al. (2010). Lessons learned about mixed methods research strategies in systems engineering: Evidence from PhD dissertations. 8th Conference on Systems Engineering Research. Hoboken, NJ.
White House (2006). The national security strategy of the United States of America. White House. Washington, DC, Government Printing Office.
Wiegers, C. (2000). Stop promising miracles. Software Development Magazine.
Yin, R. K. (2009). Case study research: Design and methods. Thousand Oaks, CA, Sage Publications.

6.2 Related Published Work

Liu, K. K., R. Valerdi et al. (2009). Economics of human systems integration: The Pratt & Whitney F119 engine. Human Systems Integration Symposium. Annapolis, MD.

Liu, K. K., R. Valerdi et al. (2009). Economics of human systems integration: A systems engineering perspective. 7th Annual Conference on Systems Engineering Research. Loughborough, UK.

Valerdi, R., K. K. Liu et al. (2010). Lessons leared about mixed methods research strategies in systems engineering: Evidence from PhD dissertations. 8th Conference on Systems Engineering Research. Hoboken, NJ.

Liu, K. K., R. Valerdi et al. (2010). Better requirements decomposition guidelines can improve cost estimation of systems engineering and human systems integration 8th Annual Conference on Systems Engineering Research. Hoboken, NJ.

Liu, K. K., R. Valerdi et al. (2010). "The F119 engine: A success story of human systems integration in acquisition." Defense Acquisition Review Journal **17**(2): 284-301.

Valerdi, R. and K. K. Liu (2010). Parametric cost estimation of human systems integration. The economics of human systems integration: Valuation of investments in people's training and education, safety and health, and work productivity. W. B. Rouse. Hoboken, NJ, John Wiley and Sons. (forthcoming)

7 APPENDICES

7.1 Workshop 1 Handouts

24th International Forum on COCOMO and Systems/Software Cost Modeling

COSYSMO Workshop

Exercise: *"COSYSMO for HSI"*

Thank you for participating in this exercise as part of the COSYSMO workshop session. Your time and input is much appreciated. Please follow along on these handouts as the presenter goes through the corresponding slides.

First, the Basics:

Name: ________________________________Em**ail**:____

Affiliation: ____

What is your familiarity with Systems Engineering?

None Some familiarity Extensive

Please describe:

What is your familiarity with Requirements Management/Engineering?

None Some familiarity Extensive

Please describe:

What is your familiarity with Requirements Decomposition?

None Some familiarity Extensive

Please describe:

What is your familiarity with the Constructive Systems Engineering Cost Model (COSYSMO)?

First exposure | Heard of it, never used | Used briefly | Used extensively | Helped to develop it

Please describe:

What is your familiarity with Human Systems Integration?

Please describe:

Part 1: Determine the System of Interest

#1: Too High | Sea-Level | Too Low
Comments:

#2: Too High | Sea-Level | Too Low
Comments:

#3: Too High | Sea-Level | Too Low
Comments:

#4: Too High | Sea-Level | Too Low
Comments:

#5: Too High | Sea-Level | Too Low
Comments:

#6: Too High | Sea-Level | Too Low
Comments:

#7: Too High | Sea-Level | Too Low
Comments:

#8: Too High | Sea-Level | Too Low
Comments:

Applying Cockburn's Use Case Hierarchy:

"Sea-level" use cases are tasks where the user "makes their money".

High:
"Deliver quality product to Client"
Sea-level:
"Develop monthly reports for client".
Low:
"Log into workstation"

Now think of the system being designed as a "user". How does the system "make its money"?

Hints:

If it is lower, what higher-level requirement could be used to capture this concept for the glass console?

If it is higher, what derived requirement(s) would capture this requirement's intent at the level of the glass console?

If you can't answer these questions, perhaps the level of interest is actually correct!

Part 2: Can the System be tested, verified, OR designed?

#1: Comments:

#2: Comments:

#3: Comments:

#4: Comments:

#5: Comments:

#6: Comments:

#7: Comments:

#8: Comments:

Hints

If you have to decompose the requirement in order to be able to test, verify, or design it, then the answer is no.

Does the requirement depend on factors outside system boundaries?

What is the test?

How would you design the requirement?

Is the requirement too vague?

Part 3: Do these Requirements have an HSI Impact on the System-of-Interest?

#1: None | One requirement | Many requirements
Comments:

#2: None | One requirement | Many requirements
Comments:

#3: None | One requirement | Many requirements
Comments:

#4: None | One requirement | Many requirements
Comments:

#5: None | One requirement | Many requirements
Comments:

#6: None | One requirement | Many requirements
Comments:

#7: None | One requirement | Many requirements
Comments:

#8: None | One requirement | Many requirements
Comments:

HSI requirements include, but are not limited to, any requirement pertaining to one or more domains of HSI, or the integration of those domains. Broadly, the term encompasses **any requirement that contributes to the integration of human considerations into the system being developed**.

Commonly recognized HSI domains:

Manpower
Personnel
Training
Human Factors Engineering
Ergonomics
Survivability
Safety
Occupational Health
Habitability
ESOH
Health Hazards
Environment

Workshop 1 Phase 1

Cautions and Warnings. Method for displaying system warnings, cautions, and alarms must be appropriate given the importance of the situation (**Threshold**).

Timeliness and Accuracy of Information. The glass console will present accurate and timely data from many important systems to the driver. The console must detect data errors and display an error message instead of incorrect information (**threshold**). An error message must be displayed if the system detects a lag of greater than (**threshold=**20ms, **objective**=2ms)

User Interface. The user must be able to interact with the console using either touch or voice (**threshold**).

Response Time. User must be able to perceive and process all information displayed by console in under (**threshold**=20ms, **objective**=2ms).

Map Readability. Maps displayed for the GPS function of the console shall be readable with 20/20 vision at 1 meter (**threshold**).

Emergency Override. In the case that the console receives an emergency interrupt from vehicle sensors, it shall display the appropriate emergency caution, alarm, or alert in under (**threshold=**2ms**)**

Data Redundancy. All sensors built into the vehicle must provide a copy of its output to the glass console for possible integration into alerts, alarms, or cautions (**threshold**).

Speech Displays. If the console is to employ computer-generated speech to communicate with the driver, speech rate must not exceed 150 wpm (**threshold**)

Workshop 1 Phase 2

Cautions and Warnings. Method for displaying system warnings, cautions, and alarms must be appropriate given the importance of the situation (**Threshold**).

Response Time. The driver must be able to perceive and process all information displayed by console in under (**threshold**=20ms, **objective**=2ms).

Display Contrast. All text displayed by the glass console must have a contrast ratio of (**objective**=5000:1, **threshold**=1000:1), defined as the ANSI standard contrast ratio in an ideal room.

Display Reliability. The glass console must not fail due to electrical malfunction within the lifetime of an average vehicle (assume 10 years) (**threshold**).

Mitigation of Divided Attention. The driver shall be able to operate all aspects of the glass console without having a measurable impact upon driving performance (**threshold**).

Console Startup/Shutdown Times. The glass console must be available for the driver to interface with within (**objective=**2s, **threshold**=5s) of the beginning of the ignition sequence. The glass console must be in a low-power or standby state within (**threshold=**10s) of vehicle power-off.

Function Execution. The driver must be able to perform any single task (menu selection, radio station switching) in under (**threshold**=5 s)

Data Integration. All displays within the glass console shall use a standard set of buttons and labels to improve intuitiveness (**threshold**)

Workshop 1 Phase 3

Timeliness and Accuracy of Information. The glass console will present accurate and timely data from many important systems to the driver. The console must detect data errors and display an error message instead of incorrect information (**threshold**). An error message must be displayed if the system detects a lag of greater than (**threshold=**20ms, **objective**=2ms)

Temperature Control. The inside temperature must remain within 2 degrees Celsius of the desired temperature set by the driver during normal driving conditions (**threshold**).

Passenger Safety. In the case of a front or side-impact collision at speed over 80 mph, but under 100mph, chance of survival for all passengers must be greater than 95%, given an average adult male or female (**threshold**).

Documentation. All manuals/documentation for the vehicle will include all information necessary to operate the vehicle in a safe and effective way (**threshold**).

Air Quality. Air quality must be maintained to standards as set forth in QLT-STD-1234 (**threshold**). In the case that air quality degrades below these standards, appropriate alarms or alerts shall warn the driver (**threshold**).

Maintenance Time. All minor to medium-level vehicle maintenance tasks shall take no longer than (**objective=**1hr, **threshold**=2hrs) to complete.

Ambient Light. All displays must function to identical performance characteristics in low light as well as direct sunlight (**threshold**).

Console Startup/Shutdown Times. The glass console must be available for the driver to interface with within (**objective=**2s, **threshold**=5s) of the beginning of the ignition sequence. The glass console must be in a low-power or standby state within (**threshold=**10s) of vehicle power-off.

7.2 Workshop 2 Handouts

COSYSMO Workshop
Exercise: "COSYSMO for HSI"

Thank you for participating in this exercise as part of the COSYSMO workshop session. Your time and input is much appreciated.

<u>First, the Basics:</u>

Name: ________________________________**Email**:____

Affiliation: ____

What is your familiarity with Systems Engineering?
First exposure | Heard of it | Trained | Experienced | Expert

What is your familiarity with Requirements Management/Engineering?
First exposure | Heard of it | Trained | Experienced | Expert

What is your familiarity with Requirements Decomposition?
First exposure | Heard of it | Trained | Experienced | Expert

What is your familiarity with the Constructive Systems Engineering Cost Model (COSYSMO)?
First exposure | Heard of it | Trained | Experienced | Expert

What is your familiarity with Human Factors/Human Factor Engineering?

First exposure | Heard of it | Trained | Experienced | Expert

What is your familiarity with Human Systems Integration?

First exposure | Heard of it | Trained | Experienced | Expert

Additional Comments:

<u>Example of a nominal requirement</u>

Threshold: Operators shall be able to read the XX display (where XX display is the system being designed) during day and night, with no visible signature at night from 10-50m. Device must enable operators to keep head up while reading data on computer.

<u>Requirements to be judged</u>

1. Threshold: When employing proper escape/evasion and camouflage/concealment techniques, the XX will have a visible signature with less than a 10% probability of detection at 328 ft (100m) with the unaided eye, optical magnification, and NV devices.
2. Threshold: Hands-on training is required for 25 personnel at the centralized weather organizations to operation and maintain XX provided systems. For external users, the contractor is required to develop computer based training modules for all external services. Objective 50 personnel
3. The maintainer will be able to complete maintenance corrective actions within (T=30, O=10) minutes
4. The operating altitudes of the XX (platform) will be above all small arms threats and smaller caliber light AAA threats. The modification package (this CDD) shall detect, provide avoidance recommendation, and/or counter all anticipated threats in a low threat environment (e.g. small arms, light AAA, & MANPADS) while maintaining a persistent, tactically responsive presence over the area of operations, day or night. Finally, the modification package will enhance the donor aircraft's survivability with standoff posture through Sensor Capabilities and SOPGM
5. The system must be designed to eliminate or mitigate safety, health or physical risks. Where hazards/risks exist, health and safety equipment and/or procedures must be identified. Health and Safety procedures and engineering design considerations must conform to AF Operational and Safety health standards. Crew task load, fatigue factors, broad range of operating environments, and data assimilation must be considered.

Example of a nominal requirement

Threshold: Operators shall be able to read the XX display (where XX display is the system being designed) during day and night, with no visible signature at night from 10-50m. Device must enable operators to keep head up while reading data on computer.

Requirements to be judged

6. Personnel with the appropriate aptitudes, physical/mental abilities will be employed for the system. In addition, human performance shall be optimized by assessing the workload and tasks and ensuring that each crew member can accomplish the mission without experience task saturation.
7. The system inspection concept shall be designed to maximize flying operations by aligning scheduled inspections/maintenance intervals with donor aircraft.
8. The system will be required to operate in all environmental regions of the globe without perceptible degradation in performance.
9. The system should have internally installed instrumentation interfaces to support testing and training that is interoperable with existing and planned test and training systems. All electronic tactical systems should have an automatic fault code recorder and tracking system to assist maintenance personnel in maintenance and repair work.
10. Personnel shall be able to utilize the system as required while wearing the Mission Oriented Protection Posture (MOPP) IV ensemble. The system must enable operators to perform tasks while in MOPP IV ensembles and perform mission tasks in an operational environment.

ANSWER SHEET – ROUND 1 (individual round)[1]

<u>Example of a nominal requirement</u>

Threshold: Operators shall be able to read the XX display (where XX display is the system being designed) during day and night, with no visible signature at night from 10-50m. Device must enable operators to keep head up while reading data on computer.

Compared to the nominal requirement, this requirement is how much more effort?

1.	1x	2x	4x	8x
2.	1x	2x	4x	8x
3.	1x	2x	4x	8x
4.	1x	2x	4x	8x
5.	1x	2x	4x	8x

6.	1x	2x	4x	8x
7.	1x	2x	4x	8x
8.	1x	2x	4x	8x
9.	1x	2x	4x	8x
10.	1x	2x	4x	8x

[1] Answer sheets for Rounds 2 and 3 were identical.

Workshop 1 Phase 1

Cautions and Warnings. Method for displaying system warnings, cautions, and alarms must be appropriate given the importance of the situation (**Threshold**).

Timeliness and Accuracy of Information. The glass console will present accurate and timely data from many important systems to the driver. The console must detect data errors and display an error message instead of incorrect information (**threshold**). An error message must be displayed if the system detects a lag of greater than (**threshold=**20ms, **objective**=2ms)

User Interface. The user must be able to interact with the console using either touch or voice (**threshold**).

Response Time. User must be able to perceive and process all information displayed by console in under (**threshold**=20ms, **objective**=2ms).

Map Readability. Maps displayed for the GPS function of the console shall be readable with 20/20 vision at 1 meter (**threshold**).

Emergency Override. In the case that the console receives an emergency interrupt from vehicle sensors, it shall display the appropriate emergency caution, alarm, or alert in under (**threshold=**2ms**)**

Data Redundancy. All sensors built into the vehicle must provide a copy of its output to the glass console for possible integration into alerts, alarms, or cautions (**threshold**).

Speech Displays. If the console is to employ computer-generated speech to communicate with the driver, speech rate must not exceed 150 wpm (**threshold**)