SANDIA REPORT

SAND2016-2607
Unlimited Release
Printed March 2016

Systematic Literature Review: How is Model-Based Systems Engineering Justified?

Edward R. Carroll and Robert J. Malins

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

Systematic Literature Review: How is Model-based Systems Engineering Justified?

Edward R. Carroll
Robert J. Malins
NW & Labs Futures Analysis

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0421

Abstract

The genesis for this systematic literature review was to search for industry case studies that could inform a decision of whether or not to support the change process, investment, training, and tools needed to implement an MBSE approach across the engineering enterprise. The question asked was, how the change from a document-based systems engineering approach (DBSE) to a model-based systems engineering approach (MBSE) is justified? The methodology employed for this systematic literature review was to conduct a document search of electronically published case studies by authors from the defense, space, and complex systems product engineering industries. The 67 case studies without metrics mainly attributed success to completeness, consistency, and communication of requirements. The 21 case studies with metrics on cost and schedule primarily attributed success to the ability of an MBSE approach to improve defect prevention strategies.

The primary conclusion is that there is a significant advantage to project performance by applying an MBSE approach. An MBSE approach made the engineering processes on a complex system development effort more efficient by improving requirements completeness, consistency, and communication. These were seen in engineering processes involved in requirements management, concept exploration, design reuse, test and qualification, Verification and Validation, and margins analyses. An MBSE approach was most effective at improving defect prevention strategies. The approach was found to enhance the capability to find defects early in the system development life cycle (SDLC), when they could be fixed with less impact and prevented rework in later phases, thus mitigating risks to cost, schedule, and mission. However, if a program only employed an MBSE approach for requirements management, advantages from finding defects early could not be leveraged in later phases, where the savings in cost and schedule from rework prevention is realized.

Significant performance success was achieved when the systems engineer (SE) held a leadership role over engineering processes. A number of the case studies addressed a general lack of skilled MBSE engineers as a major hindrance to implementing an MBSE approach successfully.

Contents

Figures

Tables

Nomenclature

BCR	Baseline Change Request – a request that changes the previously established project cost or schedule
DBSE	Document-based systems engineering – an approach to engineering systems that relies upon paper or digital textual documentation to record system specifications and other development project related information
FTE	Full-time equivalent – a labor resource consuming the equivalent of a full-time employee (40 hours/week)
INCOSE	The International Council on Systems Engineering – a professional society
MBPLM	Model-based product line management – an approach to manage production line development efforts that relies upon digital diagrams (called models)
MBSE	Model-based systems engineering – an approach to engineering systems that relies upon digital diagrams (called models) to record system specifications and interfaces to subordinate component specifications
Models	Diagrams, drawings, and databases, usually digitally rendered and electronically maintained, that utilize a specialized nomenclature to represent system specifications as objects, links, entities, attributes, relationships, processes, dataflows, actors, and states.
NPV	Net present value – the value of an investment by adding the present value of expected future cash flows to the initial cost of the investment
OEM	Original equipment manufacturer – a company that makes a part or subsystem that is used in another company's end product
OMG	Object Management Group – an industry consortium
ROI	Return-on-investment – the gain or loss from an investment of resources (financial, labor, materials)
SDLC	System development life cycle – the combined and sequential phases to engineer and produce a system. Typically phases include: concept, requirements definition, design definition, construction (or manufacturing), and tests
SDR	System Design Review – a critical review commonly included in the development life cycle of government systems
SE	Systems Engineering – the engineering process that considers the system development processes in their entirety and as they interrelate
SLC	System life cycle – the combined and sequential phases to engineer, produce, operate, and retire a system (extending beyond development)
SysML	Systems Modeling Language – A nomenclature for use in digital systems engineering models developed by OMG and adopted by INCOSE as a standard for use in an MBSE approach
V&V	Verification and Validation – the processes to verify that the system developed meets all of the design specifications and validates that the system delivered is what the customer ordered

1. Executive Summary

The genesis for this systematic literature review was to search for industry case studies that could inform a decision of whether or not to support the change process, investment, training, and tools needed to implement an MBSE approach across the engineering enterprise. The question asked was, how is the change from a document-based systems engineering approach (DBSE) to a model-based systems engineering approach (MBSE) justified? We identified relevant case studies, appraised them for quality, and drew the following conclusions from the findings.

There is a significant advantage to project performance by applying an MBSE approach. An MBSE approach made the engineering processes on a complex system development effort more efficient by improving requirements completeness, consistency, and communication. These were seen in engineering processes involved in requirements management, concept exploration, design reuse, test and qualification, Verification and Validation, and margins analyses. An MBSE approach was most effective at improving defect prevention strategies. The approach was found to enhance the capability to find defects early in the system development life cycle (SDLC), when they could be fixed with less impact and prevented rework in later phases, thus mitigating risks to cost, schedule, and mission. However, if a program only employed an MBSE approach for requirements management, advantages from finding defects early could not be leveraged in later phases, where the savings in cost and schedule from rework prevention is realized.

Significant performance success was achieved when the systems engineer (SE) held a leadership role over engineering processes. A number of the case studies addressed a general lack of skilled MBSE engineers as a major hindrance to implementing an MBSE approach successfully.

There are a number of prerequisites for any enterprise to employ an MBSE approach:

- Mature, well-documented, and enterprise-wide SE processes that span the SDLC
- Trained systems engineers in MBSE techniques
- Access to training in the SE processes for all engineers
- Defined processes for model management throughout the SDLC
- Investment in full-scale MBSE tools

In addition to these prerequisites, the enterprise would need to make the following commitments:

- Initiate modeling with appropriate staffing levels at the beginning of the program
- Configuration manage the model “first change the model, the model is the design”
- Provide continuous resources to maintain the models throughout the SDLC
- Provide MBSE resources and models to support system testing, qualification, and V&V
- Provide appropriate sustained computing infrastructure throughout the SDLC

The case studies confirm that enterprises acquired significant benefits from an SE approach in general and an MBSE approach in particular by making these investments and commitments.

2. Introduction

Although it is expected that most of the readers of this systematic literature review will be very familiar with the processes involved in systems engineering (SE) and perhaps also model-based systems engineering (MBSE), we provide the following definitions for those who may not be familiar with this topic.

What is a system?

For the purposes of this review, a system is defined as a technology that is

> "… an integrated set of elements, subsystems, or assemblies that accomplish a defined objective." [1]

What is a system life cycle?

The processes to create a system typically follow a sequence of phases, a life cycle. Therefore, the system life cycle (SLC) is the combined and sequential phases to engineer, produce, operate, and retire a system. The life cycle is referred to as the system development life cycle (SDLC) when only concerned with the creation process. A fundamental understanding of this systematic literature review is the role that a systems engineering or a model-based system engineering approach contributes to the processes to engineer a system through the life cycle. Figure 1 illustrates the main phases of a life cycle to develop a complex system, as employed by three U.S. Government agencies.

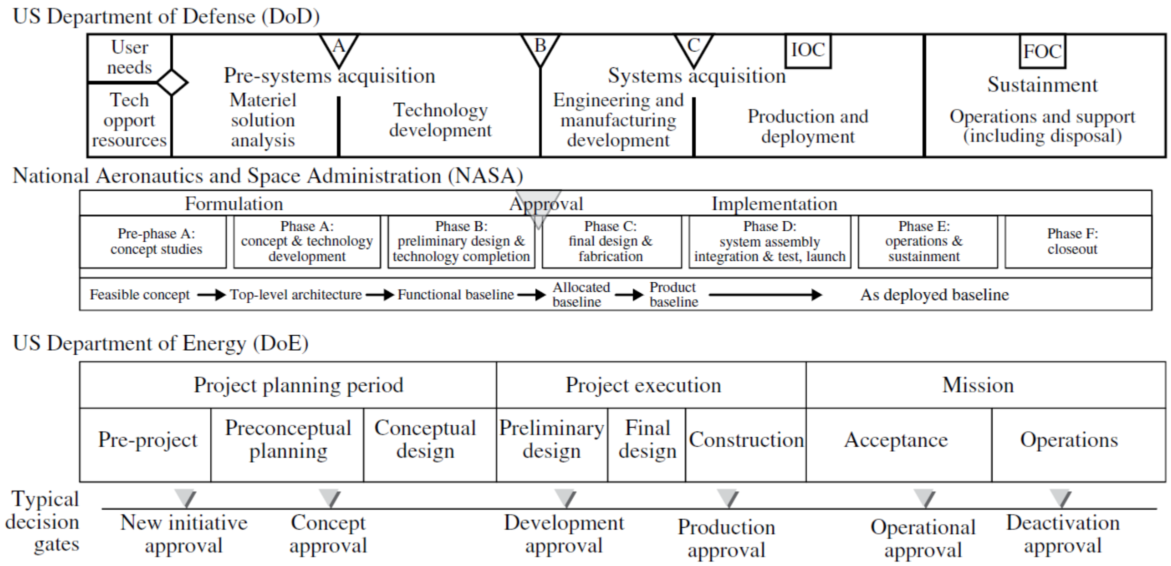

Figure 1: System Life Cycle Phases
© 2011 by K.J. Forsberg [1]

What is systems engineering?

Systems engineering is one of the multiple engineering disciplines, along with electrical, mechanical, software, or others that encompasses the processes and skillsets needed to develop a system. The International Council on Systems Engineers (INCOSE) defines the discipline of systems engineering as:

> Systems Engineering (SE) is a perspective, a process and a profession … focused on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. [1]

Systems engineering processes were defined in the 1960s and 1970s. However, SE as a separate engineering discipline grew out of the need to improve the engineering processes (around those areas defined in the quote above) during the 1990s as engineering programs became more complex, before the advent of digital modeling.

When discussing systems engineering **–** without referring to the use of engineering-models the common industry term used is either an SE approach or a document-based systems engineering (DBSE) approach. We will use both terms in this review depending upon the terminology used in the cited case studies.

What is model-based Systems Engineering (MBSE)?

> Model-based systems engineering (MBSE) is the formalized application of modeling to support systems requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. [1]

Contrasting a traditional (document-based) systems engineering approach (DBSE) with that of a model-based systems engineering approach (MBSE):

> In a document-based SE approach, there is often considerable information generated about the system that is contained in documents and other artifacts such as specifications, interface control documents, system description documents, trade studies, analysis reports, verification plans, procedures, and reports. The information contained within these documents is often difficult to maintain and synchronize, and difficult to assess in terms of its quality (correctness, completeness, and consistency). [1]
>
> In an MBSE approach, much of this information is captured in a system model or set of models. The system model is a primary artifact of the SE process. MBSE formalizes the application of SE through the use of models. [1]

Use of an MBSE tool as a central, common, and integrated repository is significant to the distinguishing difference between an SE and MBSE approaches. For example, a model can be drawn in a non-MBSE tool, e.g. Visio, but the end-result is a document-centric approach.

There has been a groundswell to use an MBSE approach, due to the reported benefits. The International Council on Systems Engineering (INCOSE) describes the benefits of an MBSE approach as:

- Improved communications
- Increased ability to manage system complexity
- Improved product quality
- Enhanced knowledge capture
- Improved ability to teach and learn SE fundamentals [1]

However, as a counterweight to these described benefits, there are cost and difficulties toward implementing an MBSE approach that need to be overcome. For example, an MBSE approach is significantly different from a DBSE approach, forcing changes to engineering processes, tools, and communication methods in order to be successful, and requiring a substantial financial investment for training and tooling. Changing how information is communicated (using models, instead of simple text) is a difficult process in itself that involves all stakeholders, which on a large complex system project could be thousands of people. There are technical issues around application, models, data standards, security, and information configuration management that need to be resolved at the start of any program. In addition, an MBSE approach can change the labor distribution curve. Instead of a flat-line deployment of SE resources across the system development life cycle, an MBSE approach can emphasize a greater use of SE resources early in the life cycle, forcing cost expenditures earlier than in a DBSE approach. This is because MBSE tools will typically include functionality that focuses more effort on completing requirements and interfaces throughout the early phases.

Therefore, the genesis for this systematic literature review was to search for industry case studies that might validate or disprove the reported benefits, preferably with quantifiable metrics that could inform a decision of whether or not to support the change process, investment, training, and tools needed to implement an MBSE approach across the engineering enterprise.

2.1. Framing the Question

- How is the change from a DBSE approach to a MBSE approach justified?
    - ***The Population***: Those systems/programs/projects using SE processes within the defense, space, and complex system product engineering industries.
    - ***Study Design***: Case studies of any design examining the justification for implementing (or not implementing) a MBSE approach were drawn from the population groups.
    - ***The Outcomes***: System/program/project performance as defined by industry standards – cost savings, schedule performance, and/or defect rate.
    - ***The Comparisons***: Conceptually, projects using a DBSE approach were compared against projects using an MBSE approach. However, because the cost to build two

highly complex systems is prohibitively expensive, controlled studies of side-by-side comparison are not feasible. As a result, this systematic literature review relied upon one-sided, post-activity progress comparisons limited to the documented justifications from within the case studies.

2.2. Identifying the Relevant Publications

A wide range of electronic documents were sought from defense, space, and the product engineering industry to capture as many relevant case studies as possible. Case studies about systems, programs, and projects were sought from multiple industry, association, and government sources. Electronic searches were supplemented by contacting persons of known involvement in justifying an MBSE approach. Case studies were evaluated (and the selection narrowed) for those studies that defined the justification for changing to an MBSE approach. We limited our search to those case studies dating back to 2005 (the approximate starting date of digital MBSE in industry) providing a 10-year window for comparison. Exceptions to the 2005 limit were included when cited by more recent case studies.

The initial search criterion was for documents with keywords of “Model-Based Systems Engineering” OR “Systems Engineering.” This effort resulted in over 20,000 case studies. A secondary search criterion was added using AND “ROI”, AND “Justif”, which yielded 1,000 case studies from which relevant studies were selected for review. Potential relevance was examined based on a reading of the document titles and abstracts. Of this subset 865 case studies were excluded because they did not discuss justification. At this point 47 case studies were removed from the selection list as being redundant, or upon further reading were excluded because they did not justify their use of an MBSE approach. Some of these were case studies documenting return-on-investment (ROI) methodology or implementation processes.

From the remaining 88 selected case studies, 67 were separated out as those case studies that justified the use of an MBSE approach by stating the generally understood benefits of an MBSE approach, without documenting measurable metrics or results. These 67 articles came from 8 different countries, 10 from defense, 33 from aerospace-space, 5 from non-defense government applications, 6 from commercial enterprises, and 12 were academic papers.

The remaining 21 case studies were selected because they defined how the investment in an SE or MBSE approach was justified using quantifiable metrics, and were reviewed in detail. They came from 4 countries, 12 from defense, 5 from space, and 4 from commercial enterprise, with 6 of these published as an academic treatise. 9 case studies justified an SE approach without specifying the use of models and 12 used an MBSE approach. 6 case studies used an MBSE approach to develop a complex weapon system.

2.3. Assessing Study Quality

Design threshold for study selection: Selected studies were subjected to a refined quality assessment by critical appraisal of the quantitative approaches taken to justify either a DBSE or MBSE approach. These detailed quality assessments were used for exploring heterogeneity, informing decisions regarding suitability of meta-analysis, assessing the strength of inferences, and in making recommendations for future research.

Quality assessment of MBSE justification studies: After studies of an acceptable design were selected, an in-depth assessment for the risk of various biases allowed us to gauge the quality of the evidence in a more refined way. All case studies selected reported on complex technical system development projects. However, as is common with engineering case studies, particularly those submitted as presentations to industry conferences by competent practitioners, many of the case studies reviewed presented a one-sided success story. Therefore, it can be assumed that case-study author biases either exaggerated or underestimated the true ROI. No case studies were found in our literature search documenting quantifiable metrics that compared failures of either the DBSE or MBSE approaches. A ratio of 12:21 case studies reported an MBSE approach as an improvement over a DBSE approach. However, none of these case studies documented a controlled experiment comparing the two approaches side-by-side. The closest to a side-by-side comparison conducted was a single case study by Frantz [23] at The Boeing Company (from 1995, well before MBSE) and referred to by Honour [16]. Therefore, it should be recognized that all of the case study authors included in this review can be considered as overly optimistic in their claims.

In addressing our own bias, the authors of this systematic literature review acknowledge that we each have over 25 years in SE and engineering approaches employing digital modeling tools. While we both favor MBSE as an approach, we have attempted to limit our personal biases in this review through careful consideration of our analysis and peer reviews, but we identify the potential.

Comparison Limitation: It is generally understood that an MBSE approach requires a sizable initial or up-front investment, in contrast to the perceived use of a more traditional DBSE approach. The cost to build highly complex defense and space systems twice, just to make a side-by-side comparison is prohibitively expensive. As a result, this systematic review will rely upon comparisons limited to the documented justifications from within the case studies. The general low statistical quality of the studies means that the results must be interpreted with caution and further study is warranted to validate the claims contained in these case studies.

2.4. Summarizing the Evidence

Eighty-eight case studies were reviewed. Of these, 67 case studies reported justifying their investment in an MBSE approach with unquantified value statements in the following manner (values overlap):

- 35 (52%) reported generic claims of program improvement
- 34 (51%) reported generic claims of technical improvement
- 9 (13%) reported claims to improve control of complexity
- 16 (24%) reported claims to improve communication
- 16 (24%) reported claims to ensure consistency
- 16 (24%) reported claims to ensure completeness of requirements
- 14 (21%) reported claims to ensure completeness of other design aspects
- 4 (6%) reported claims to maintain currency in model artifacts
- 7 (10%) reported claims to enable re-use of designs and design information
- 5 (7%) reported claims to improve ability to address stakeholder diversity

Of the 88 case studies reviewed, 21 case studies justified the use of either an SE or MBSE approach with quantifiable metrics in the following manner:

- 10 (47%) reported metrics that illustrate reductions in defect rates or in preventing rework
- 8 (38%) reported metrics that illustrate reductions in cost and schedule
- 3 (14%) reported metrics that illustrate control of requirements, complexity or risk (without defining the corresponding impact on either defects or costs)
- 4 (19%) were published outside of the USA
- 12 (57%) reported on defense systems
- 5 (24%) reported on space systems
- 4 (19%) reported on commercial systems
- 6 (29%) were published as an academic treatise
- 9 (43%) reported on a systems engineering approach without specifying the use of models
- 12 (57%) reported on an MBSE approach
- 6 (29%) reported on an MBSE approach used to develop a complex weapon system

3. Interpreting the Finding

There have been arguments made in industry and academia that justifying a change in an engineering approach should not be attempted, because there are valid reasons why success within one organization is not transferrable to another. Sheard and Miller [2], in their article, “The Shangri-La of ROI,” described the appropriate questions as follows:

> “Any estimation model must be examined to determine whether the numbers can be replicated in a specific situation.
>
> - How much waste was in the baseline process of the companies surveyed? How much waste is in your baseline process?
> - How much of their way of doing business depended on “smart people” and how much does yours? Are your people just as smart? Smarter?
> - What did the surveyed companies consider “productivity” to be, and how does it compare to your definition?
> - How are overtime, overhead, and G&A costs considered in the definition of systems engineering costs?
> - How did the surveyed companies calculate return on investment related to a decrease in time-to market? Would such calculations apply as well in your marketplace?
> - Do they follow procurement rules similar to yours, or do you have constraints that would keep you from achieving the same improvements?” [2]

In spite of Sheard and Miller’s concerns, many projects have moved forward with the implementation of both SE and MBSE approaches in the ensuing 15 years from when these comments were made. The case studies reviewed in this document provide rich examples of how the authors justified making the changes to their engineering processes. Although controlled studies that would confirm a ROI expectation are difficult to perform on engineering processes for developing complex systems, lessons can still be learned from a review of case studies from across different industries, as long as one maintains a level of understanding of the differences between these examples and one’s own situation.

3.1. Key Findings from Case Studies without Metrics

The case studies that did not include quantifiable metrics primarily addressed the process by which an MBSE approach could be integrated into a complex system development effort. The most commonly discussed application of an MBSE approach was for requirements development and requirements engineering.

The most commonly quoted benefits were

- ***Completeness*** **–** more thorough analysis of the mission to form a more complete requirements set [3]
- ***Consistency*** **–** a single source of information for requirements [4], and
- ***Communication*** **–** across design teams, engineers had an improved understanding of the source of requirements and the dependencies among them [5] and [6].

Other areas where benefits were reported from an MBSE approach included:

- ***Test and evaluation*** **–** some authors reported that use of an MBSE approach enabled test planning to begin earlier in a program, improving the traceability between detailed test plans and system requirements, as well as system performance uncertainty reduction goals. [7]
- ***Verification and validation (V&V)*** **–** organizations found that consistent use of an MBSE approach in the early phases of a complex program enabled verification methods to be specified early in the program, thus enabling more thorough planning and providing detailed traceability between requirements and test plans [8] and [9].
- ***Concept exploration*** **–** developing mission architectures using an MBSE approach enabled some organizations to explore a much broader set of design options in the same amount of time and resources as conventional methods. [10]
- ***Design reuse*** **–** organizations have found that MBSE models can be reused across product lines to achieve significant savings in upfront design effort. [11]
- ***System margins analyses*** **–** detailed representation of the system in the MBSE model enables improved systems-level characterizations such as weight, budget, and power requirements analyses. [12]

A number of these case studies also addressed the challenges experienced in transitioning from DBSE to MBSE. Numerous authors cited a general lack of skilled systems engineers, and skilled MBSE engineers in particular, as a major hindrance to implementing an MBSE approach. There appears to be an overall need to develop and mentor staff skilled in MBSE tools and techniques and in the methods for employing these within a complex system development effort. Among the additional challenges described were the following:

- ***Broader adoption of SE modeling tools*** [13] **–** design engineers of all disciplines need a basic understanding of an MBSE approach in order to fully utilize the information generated by systems engineers.

- ***Development of model management processes*** [14] – life cycle management tools for MBSE model management are still limited; the model must be managed, configuration controlled, and kept up to date if it is to provide benefit to later phases of the program.
- ***Cultural barriers across the design team and stakeholder team*** [15] – both engineers and stakeholders are accustomed to reviewing documents rather than MBSE model artifacts, and in some cases will argue that "this is the way we have always don it". The process of "first change the model, the model is the design" meets with cultural resistance, and represents one of the fundamental challenges to transition from a DBSE to an MBSE approach.

3.2. Key Findings from Case Studies with Quantifiable Metrics

We have broken this section of our review findings into three subsections: those case studies that document the baseline need for changing engineering processes, those case studies that use cost and schedule to document their justification, and those that use defect correction or rework prevention to document their justification. In each subsection we start with case studies focused on an SE approach and follow with case studies focused on an MBSE approach. We do this based on the presumption that an MBSE approach is a refinement of an SE approach and the justifications for an SE approach are inherited by an MBSE approach. Some authors validate this presumption by documenting their gains from implementing an MBSE approach above the gains they received by implementing an SE approach. Please note that all of the figures included in this review are from case studies and provided only to illustrate a particular point. To more fully understand an illustration may require reading the originating case study.

3.2.1. Establishing a Baseline for Justification by Cost and Effort

Seven authors documented the baseline concepts for comparing why they felt changes to their engineering processes were necessary. Honour [16] and Mornas et al. [17] emphasized that the SE effort is more than just managing requirements and extends throughout the system development life cycle. Elm and Goldenson [18] defined the influence that SE capability maturity has on program performance. Bone and Cloutier [19] explained that models produced in an MBSE approach were broadly accepted by other engineering disciplines in their case study. Dallosta and Simcik[20] defined the basis for why an MBSE approach can reduce cost. And, Tonnellier and Terrien [21] and Saunders [22] defined the basis for why an MBSE approach can prevent rework.

3.2.1.1. Systems Engineering Effort Extends Through the Development Life Cycle

All of the 21 case studies with quantifiable metrics in this review describe the systems engineering effort as a process that extends throughout the system life cycle. This was in contrast to the 67 case studies without metrics, which most commonly discussed the application of an MBSE approach for requirements engineering. Honour [16], in his thesis, "Systems Engineering Return on Investment," provided the graphic in Figure 2 to illustrate this point by mapping the difference between the total effort and SE effort. Note that the total systems engineering effort extends through the full development life cycle. The shaded areas represent the requirements effort (what NASA calls definition).

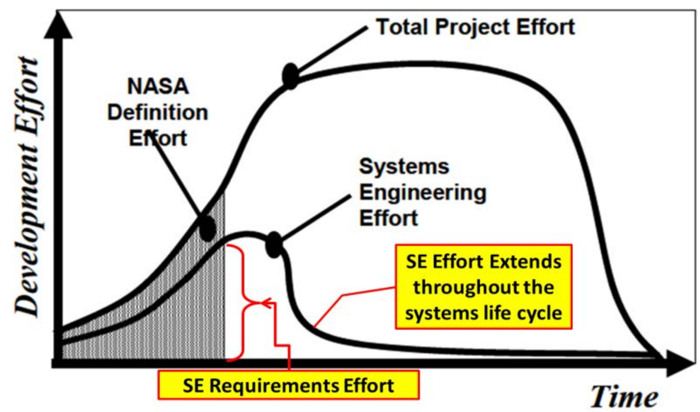

Figure 2: Comparison of SE Effort to Total Effort
Adapted with permission from E. Honour [16]

Mornas et al. [17], in their case study, "Development of Systems Engineering People to Support Major Transformation Plans in Thales (Process, Roles, Methodology & related tools)," further detailed how the SE effort is distributed across all phases of the systems development life cycle. The authors reference Figure 3, which shows the amount of SE work performed during the utilization and support phase continuing into the system retirement phase. The MBSE modeling languages and tools assume that all of these SE tasks are performed, encompassing much more than requirements management.

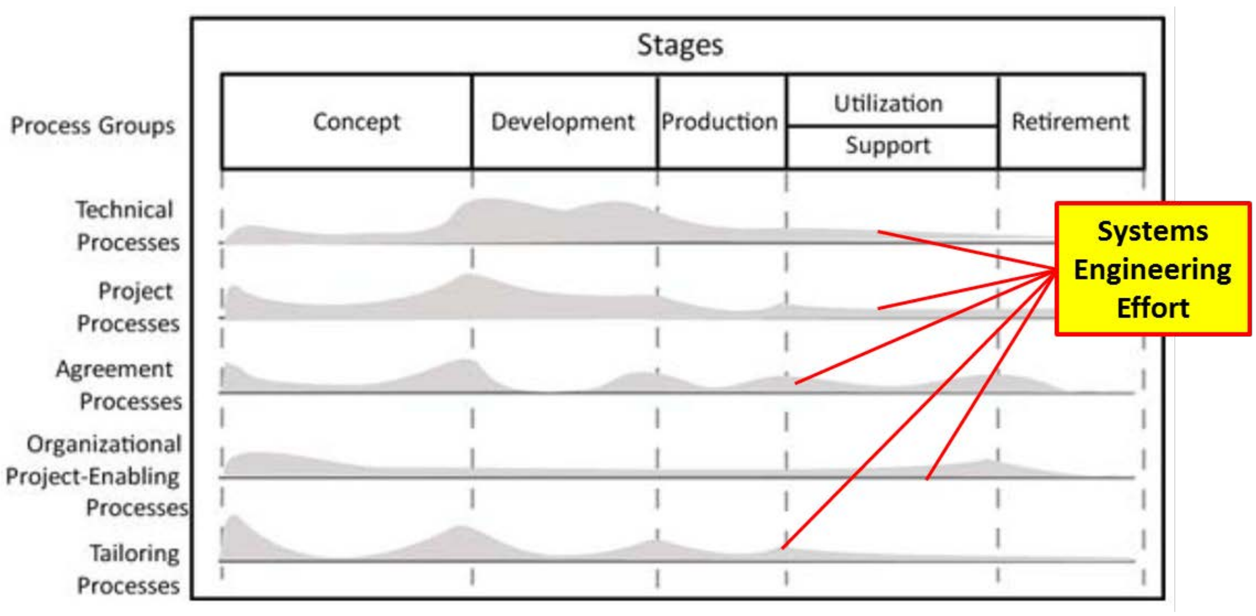

Figure 3: Systems Engineering Effort Across Project Phases
© 2011 International Council on Systems Engineering [37]

This concept is relevant to the question of how to justify an MBSE approach, as will be presented in Section 2.2.3, where 11 case studies justify an MBSE approach by cost reductions and schedule improvements attributed to the capability to find and fix defects early in a system life cycle and prevent costly rework later in the program.

3.2.1.2. Project Performance is Higher with Higher Capable Systems Engineers

Elm and Goldenson [18], in their research published in, "The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey," documented a significant correlation between SE capability and overall project performance. A dramatic 57% of projects indicated higher project performance when utilizing higher SE capability (Figure 4).

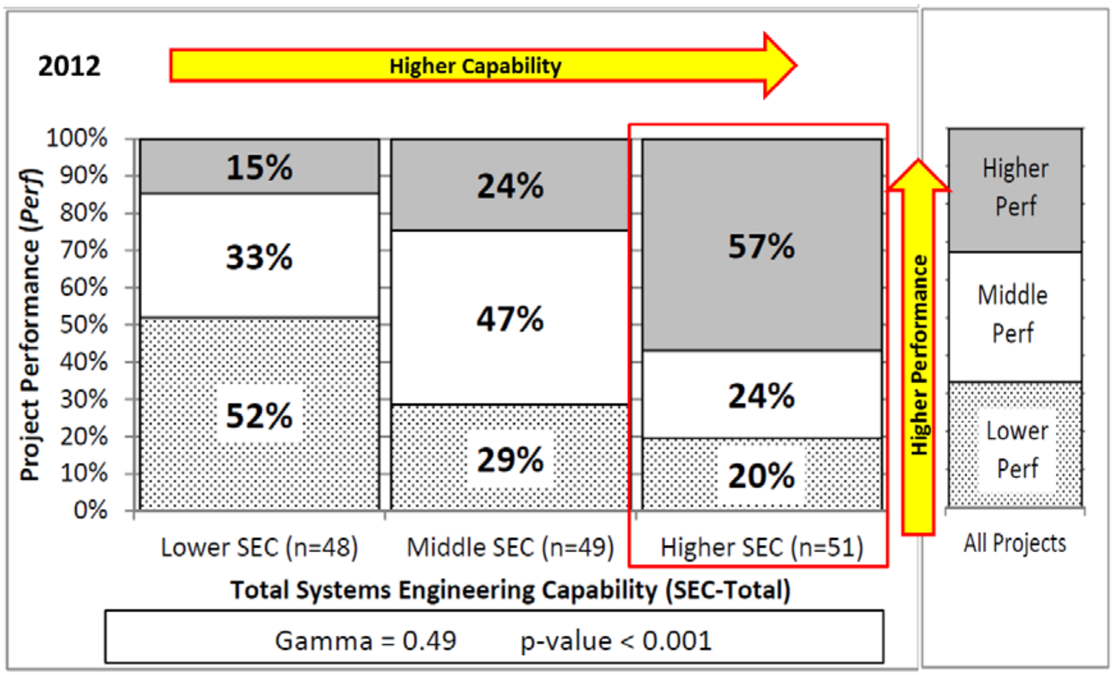

Figure 4: Systems Engineering Capability vs. Project Performance
© 2012 Carnegie Mellon University - Published in the Public Domain. [18]

This concept is relevant to the question of how to justify an MBSE approach, because those projects assessed at higher performance and capability in Elm and Goldenson's study applied SE processes across the full scope of the SDLC, as illustrated by Mornas et al. [17]. Programs employing an SE approach only for requirements, for example, rated in the low performance/low SE capability category. The authors did not separate out projects that applied an MBSE approach, however by extension, this argument can be applied to the implementation of MBSE as well as SE in general. Thus, Figure 4 further illustrates that the greatest improvements or ROI are achieved when an SE or an MBSE approach is applied throughout the system development life cycle.

3.2.1.3. Engineering Disciplines Involved in Modeling with SysML

Bone and Cloutier [19] justified the use of an MBSE approach in their case study, "The Current State of Model Based Systems Engineering: Results from the OMG™ SysML Request for Information 2009," by documenting which engineering disciplines were involved in modeling with MBSE notation (SysML). Figure 5 indicates a broad acceptance of an MBSE approach across the engineering disciplines in this case study. Note that this data contradicts the statement by Góngora et al. [11] The authors of both case studies affirm that the more engineers become familiar with models in their own disciplines, the more they will expect models from other interfacing disciplines and processes.

This concept is relevant to the question of how to justify an MBSE approach, because it reflects the growing level of acceptance with MBSE-related modeling amongst other engineering disciplines. All of the case studies reviewed reported using models to manage integration points between requirements, processes, objects, components, and with other engineering models. Requirements tracing analysis of integration points across models and margin analyses automated by MBSE tools are key processes in finding defects early.

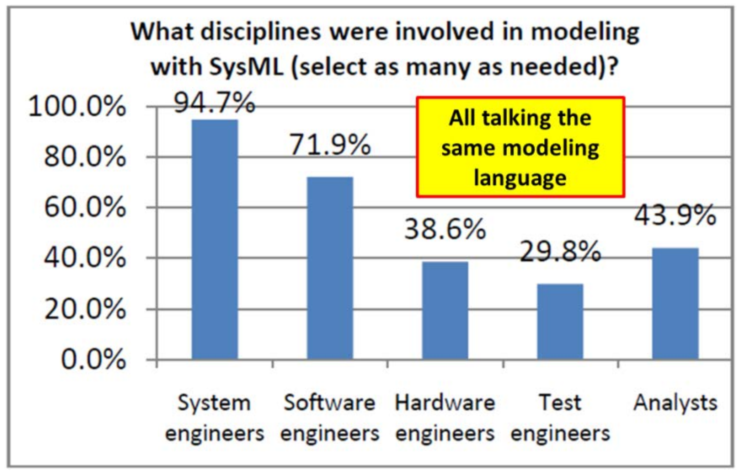

Figure 5: Engineering Disciplines Using SysML Models
Adapted with permission from M. Bone and R. Cloutier [19]

3.2.1.4. Cost Committed vs. Cost Expended

Dallosta and Simick [20] in their case study, "Driving Reliability, Availability, and Maintainability in While Driving Cost Out," illustrate in Figure 6 that up to 80% of a system life cycle cost (total ownership cost) is accrued during the operation and support phases. Recall that in Figure 3, Mornas et al. [17] illustrated that SE tasks extended throughout the operation and support phases of the system life cycle. Dallosta and Simcik, along with authors for 11 other case studies, justified an SE approach by documenting savings or cost avoidance due to reduced rework by finding and fixing defects early in the development life cycle. Those savings are often not realized until the operation and support phases.

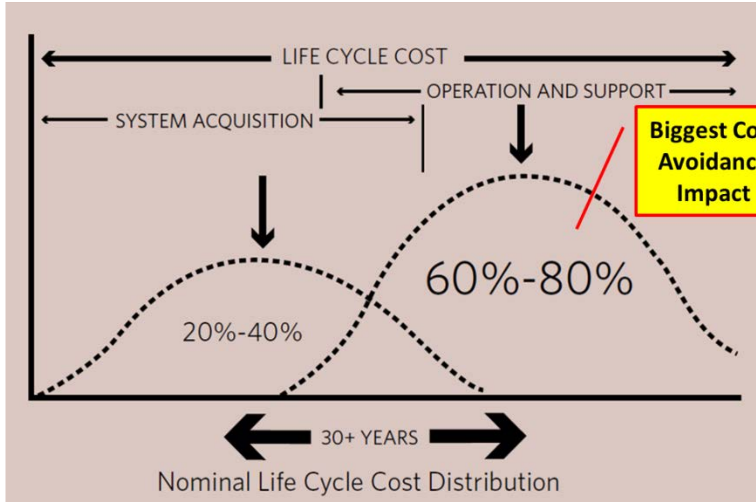

Figure 7: Cost Commitment across the SDLC
***© 2012 Defense AT&L* Magazine - *Published in the Public Domain* [20]**

In Figure 7, Dallosta and Simcik use a common representation of total life cycle cost to emphasize how important cost containment is in defense system programs. As the graph shows, 85% of project/program life cycle cost is committed by the end of system definition. The authors described how finding defects early and reducing or eliminating rework in later phases justifies an MBSE approach by maintaining or reducing cost to within committed limits.

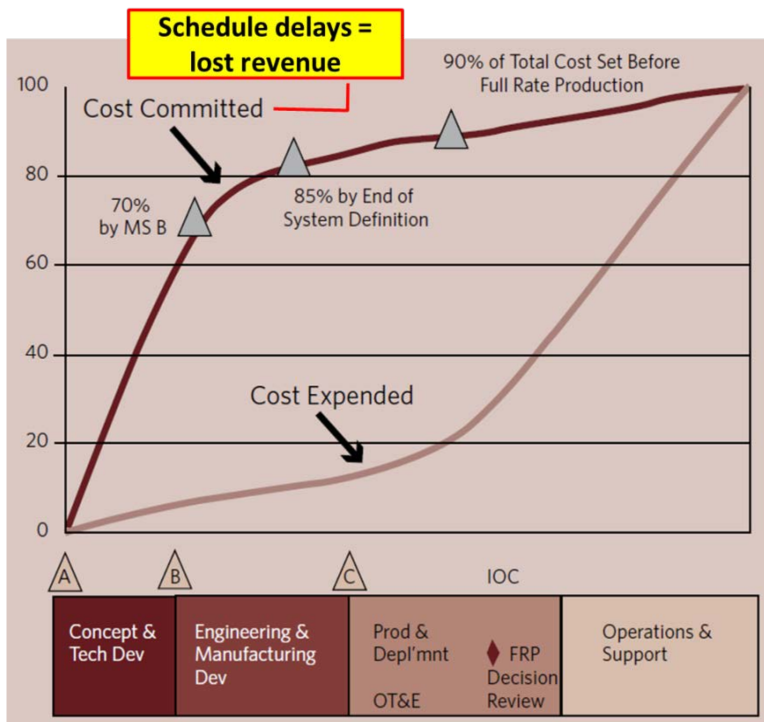

Figure 6: Normal Life Cycle Cost Distribution
© 2012 Defense AT&L* Magazine - *Published in the Public Domain [20]

This concept is relevant to the question of how to justify an MBSE approach, because it illustrates the constrained cost and schedule environment within which most system projects must operate. This sharply contrasts with the less constrained environment that commercial programs operate within, where ROI is based on the number of units sold above a breakeven point.

3.2.1.5. Defects and Rework Causes Schedule Delays

Tonnellier and Terrien [21] in their case study, "Rework: models and metrics, An Experience Report at Thales Airborne Systems," justified an MBSE approach by illustrating the impact of rework on schedule. Figure 8 shows that there are several steps in the process to resolve a defect. The central column shows the amount of effort (labor) in each step to discover and resolve a defect. The authors mention that this impact is easily tracked with MBSE tools. The far right column simply equates the effort to fix defects to rework. The authors emphasize that the level of impact increases as the complexity of a problem increases, which naturally happens as the number of interfaces increases or the system becomes more complex.

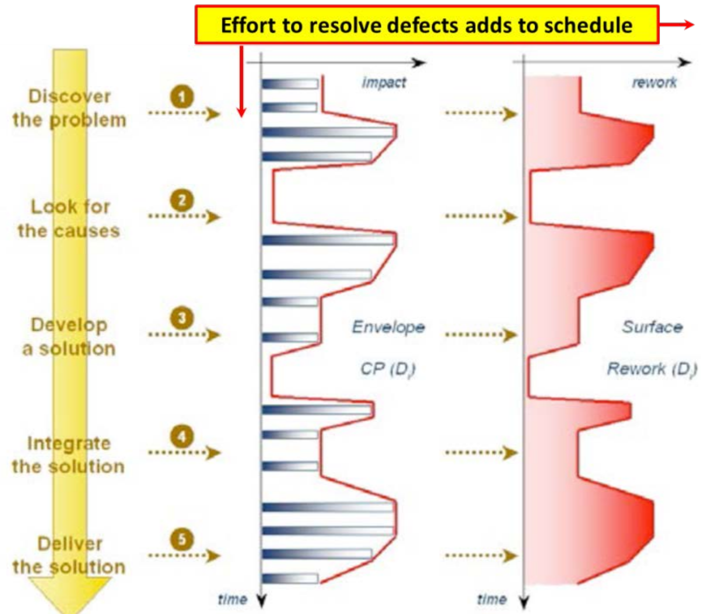

Figure 8: Rework as a Mathematical Function
© 2012 by Edmond Tonnellier and Olivier Terrien [21]

Saunders [22], in his case study, "Does a Model Based Systems Engineering Approach Provide Real Program Savings? – Lessons Learnt," justified his MBSE approach by further emphasizing the importance of finding and fixing defects as early in a program schedule as possible. He referenced an often-quoted metric from the Defense Acquisition University (1993) in Figure 9 that contrasts the cost to correct defects to the curve of committed funding, illustrating the cumulative cost of correcting defects (rework) over the lifecycle of a defense system project/program.

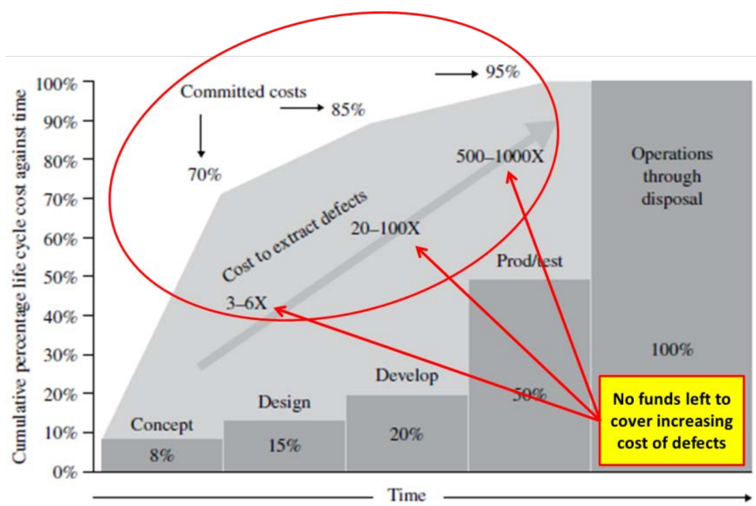

Figure 9: Cost to Fix Defect vs. Committed Cost
© 2011 Defense AT&L Magazine - Published in the Public Domain [22]

These case studies are relevant to the question of how to justify an MBSE approach, because they illustrate how fixing defects early saves cost and preserves schedule. A ratio of 14:21 case studies reported that rework late in the development life cycle possess a significant risk to the project/program for cost and/or schedule overrun. With total program cost committed in the early phases of the system life cycle, there is little room to absorb rework in the later phases. Most of the case studies in this review assume a cost commitment model similar to Figure 9.

3.2.2. Reducing Cost and Improving Schedule

Four authors justified changing processes toward SE or MBSE approaches through cost savings or schedule improvements. Honour [16 and 24] authored perhaps the most detailed analysis on justifying an SE approach based on cost savings and improved schedule performance. Frantz [23] authored perhaps the most relevant example of improvement in project schedule performance, due to implementing an SE approach. Tommasi and Vacca [25] illustrated the incremental improvements in cost and schedule performance by implementing an MBSE approach. And, Sweetman [26] described the dramatic achievements possible by modeling everything.

3.2.2.1. A Side-by-Side Comparison of Improvement from Systems Engineering

No case studies were found in this literature review that compared an MBSE approach side-by-side with a DBSE approach in a controlled experiment. The closest to a side-by-side comparison found in our search was a single case study referred to by Honour [16] in his thesis, “Systems Engineering Return on Investment,” conducted by The Boeing Company in 1995 – well before MBSE. The case study referenced by Honour was authored by Frantz [23] and gave an example of how the Boeing Company justified an SE approach by comparing improvements gained from employing three various levels of systems engineering processes on three similar projects conducted simultaneously. Honour summarizes the case study below and illustrates the performance between the three projects in Figure 10:

> A unique opportunity occurred at Boeing in which three roughly similar systems were built at the same time using different levels of systems engineering. The three systems were Universal Holding Fixtures (UHF) used for manipulating large assemblies during the manufacture of airplanes. Each UHF was of a size on the order of 10’ x 40’, with accuracy on the order of thousands of an inch. The three varied in their complexity, with differences in the numbers and types of sensors and interfaces. The three similar projects were run in parallel. Each had varying degrees of systems engineering (SE) disciplines implemented – from nearly none to high. The two projects using SE were delivered more than twice as fast. The project using the highest level of SE was delivered nearly three times faster and had the highest quality. [16]

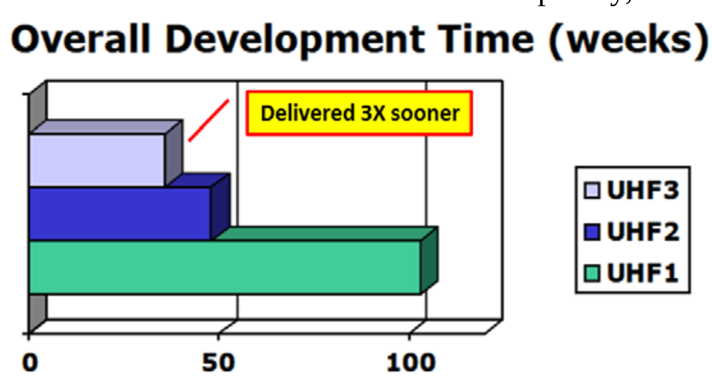

Figure 10: Three System Engineering Efforts Compared
Adapted with permission from E. Honour [16]

Frantz documented his findings or differences between the three projects described by Honour in his case study, "The Impact of Systems Engineering on Quality and Schedule *Empirical Evidence*." Note in Table 1 that in this justification only low to medium SE skills were claimed, implying that even greater improvement might be possible. The key differences between the two efforts that employed an SE approach were that the group with the best performance paid attention to the systems management discipline, updated and followed all specifications, and paid attention to external input (red boxes).

Project Trait	**UHF 1**	**UHF 2**	**UHF 3**
Systems Management experience levels	***Low***	***Low*** to ***Medium***, (relative to active NCOSE membership experience levels)	
Subcontractor Approach	***Periodic design reviews***	***Full-time Systems Engineer on*** site of major subcontractor	
Access to systems management disciplines and support	***Low*** access	***High*** access, but ***paid little attention***	***High*** access and ***paid high attention***
Systems Engineering approach for requirements	***Token requirements***	***Complete, detailed, integrated requirements*** for Robot and UHF. Developed and ***written by multi-organizational team of customers.***	
Systems Engineering approach for design	Good ***hardware and software specifications*** using multi-organizational approach.	***Functional Specification driven by the Requirements Specification*** using multi-organizational coordination, input, and reviews to over 50 people. Functional Specification fully ***addressed hardware, software, processes, and interfaces.*** The Procurement Specification fully addressed general requirements.	
Adherence to Functional Specification	Design documents took precedence. ***Specifications not followed*** and adhered to in all aspects of design and build ***but updated as required by design.***		All ***specifications updated*** as design matured. ***Specifications were followed*** and adhered to in all aspects of design and build. Controlled by formal Change Board.
Design review approach	***Weekly team reviews.*** Internal & external concerns drove smaller working meetings.	***Formal internal. Paid little attention*** to external input.	***Formal internal and external. Paid moderate attention*** to input.
Unit/Integration Test Approach	***Patterned after design***	***Driven by Functional Specification*** and ***defined early*** in the project life cycle.	
System Acceptance Test approach	Tests were defined in high-level plan.	Formal Tests were based directly from Requirements Specification Acceptance Criteria and Functional Specifications.	
Total Project Duration[4]	104 weeks	48 weeks	36 weeks
Requirements through Request for Proposal	**25 weeks**	**10 weeks--about 2.5 times faster than typical.**	
Design to production ready	52 weeks	**30 weeks**	**20 weeks--about 2.2 times faster than typical.**
Integration Test	**16 weeks**	Not applicable	**10 weeks**

Table 1: Assessment of Three System Engineering Efforts
© 1995 The Boeing Company [23]

This case study shows that when an SE approach is applied rigorously, greater improvement is achieved.

3.2.2.2. Cost Improvement from Implementing Systems Engineering

Another example described by Honour [16] in his thesis was how IBM justified employing an SE approach by tracking cost performance on eight projects.

> IBM Commercial Products division implemented new SE processes in their development of commercial software. While performing this implementation, they tracked the effectiveness of the change through metrics of productivity. Productivity metrics existed prior to the implementation and were used in cost estimation. These metrics were based on the cost per arbitrary 'point' assigned as a part of system architecting. During the SE implementation, the actual costs of eight projects were tracked against the original estimates of 'points.' Three projects used prior 'non-SE' methods, while the remaining five used the new SE methods. In the reported analysis, the data indicated that the use of SE processes improved overall project productivity when effectively combined with the project management and test processes. [16]

Year	Project	"Points"	Cost ($K)	SE Costs (%)	$/ Point
2000	Project 1	12,934	18,191	0	1,406
2000	Project 2	1,223	2,400	0	1,962
2001	Project 3	10,209	11,596	9.2	1,136
2001	Project 4	8,707	10,266	0	1,179
2001	Project 5	4,678	5,099	10.7	1,090
2002	Project 6	5,743	5,626	14.4	980
2002	Project 7	14,417	10,026	10.2	695
2002	Project 8	929	1,600	16.0	1,739

Table 2: Comparison of Eight System Engineering Projects by Function Point
Adapted with permission from E. Honour [16]

Note in Table 2, that the three projects with zero SE costs had the highest cost per point values (red box) and those projects with the highest SE cost had the lowest cost per point values, with the exception of Project 8. The exception with Project 8 is attributed to the small project size, and an indication that the project was a design project, not a full-scale development effort. This case study shows an advantage between non-SE and applied SE approaches. In addition, it provides a good example on how a common denominator can be derived to compare projects with dissimilar architectures, compensating for differences in complexity, risk and uncertainty.

3.2.2.3. Optimal Cost Performance is Achieved when SE Effort is Between 12 and 17%

A significant portion of SE effort at NASA is dedicated to requirements management, as illustrated by Honour in Figure 2. In justifying an SE approach, Honour [16] also documented a correlation between the effort spent defining requirements and project overruns on complex systems projects (Figure 11).

> The NASA data compares project cost overrun with the amount spent during phases A and B of the NASA five-phase process. The data shows that expending greater funds in the project definition results in significantly less cost overrun during project development. [16]

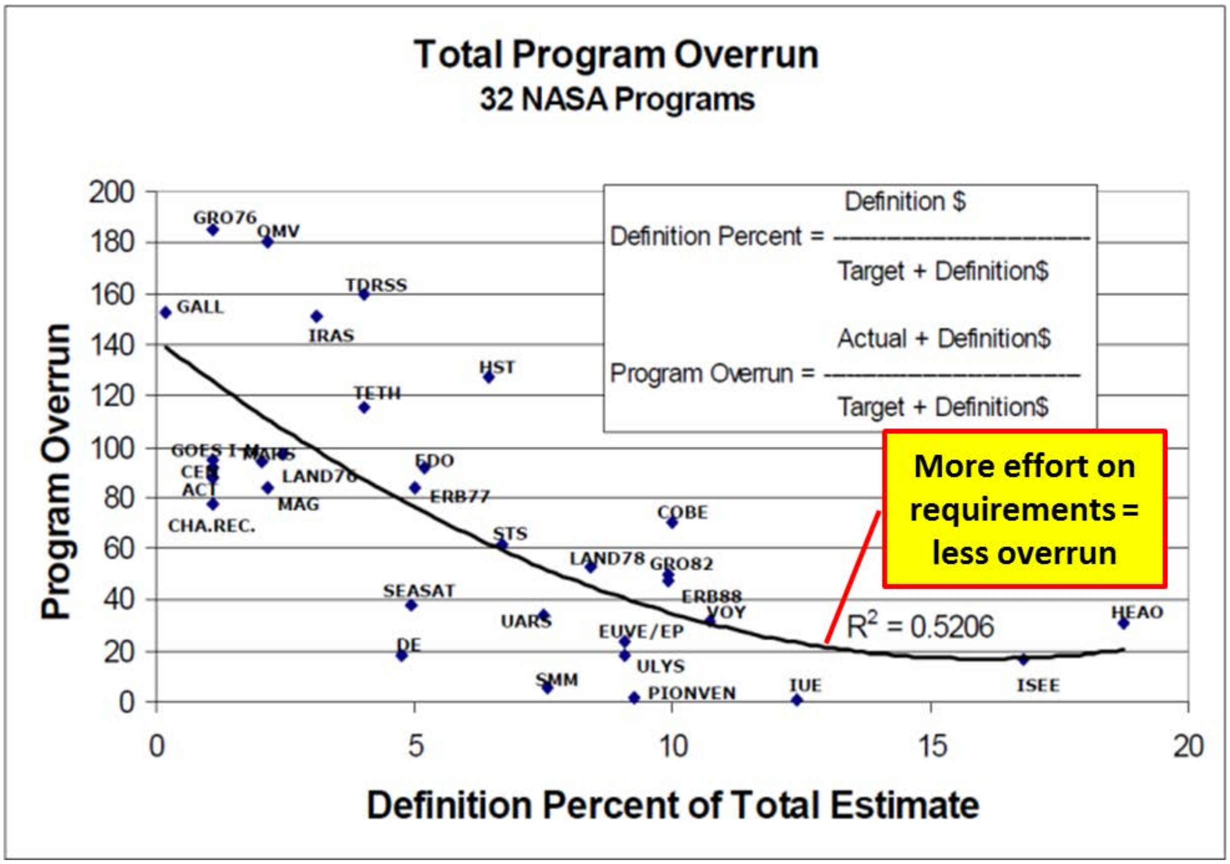

Figure 11: Total NASA Program Overrun
Adapted with permission from E. Honour [16]

Honour [24] also correlated actual/planned cost compliance against SE effort (the latest updated versions are available on Honour's website) documenting a ROI for an SE approach when an SE approach is increased to between 12% and 17% of total effort. As illustrated in Figure 12, the y-axis represents cost at 1.0, with cost overruns at increments above 1.0 and underruns below. Note two items of interest: 1) as the percent of SE effort increases, cost compliance improves; and 2) there is a point when the cost no longer improves with more SE effort.

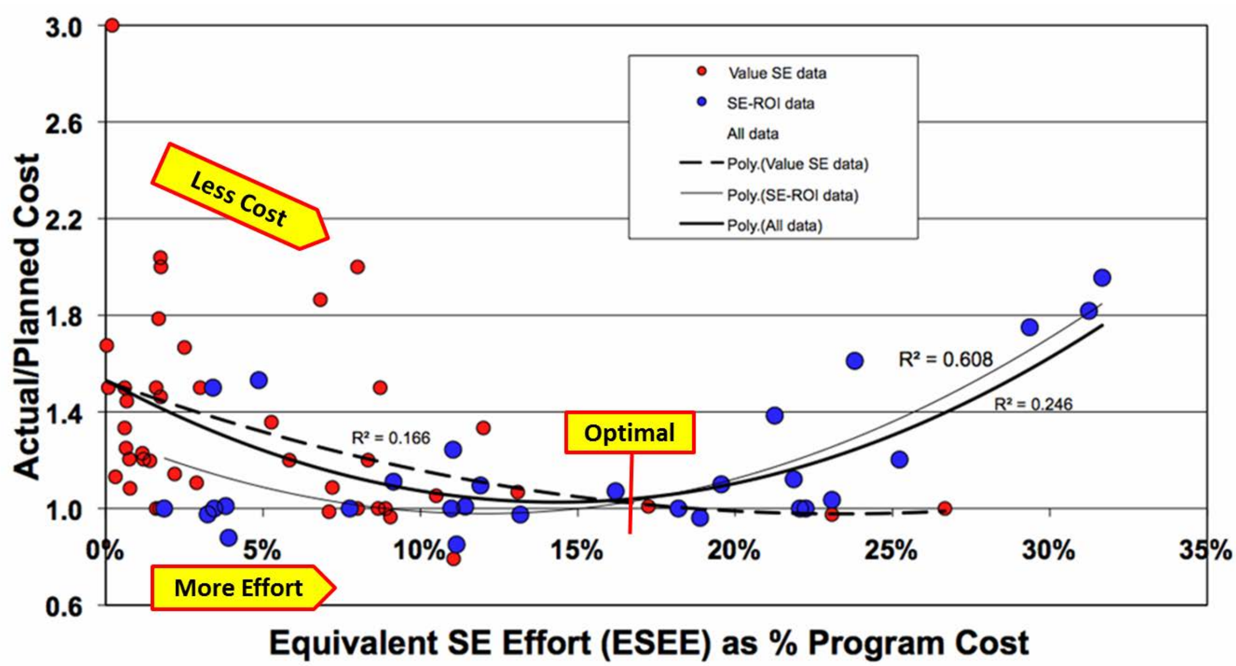

Figure 12: Systems Engineering Effort vs. Cost
Adapted with permission from E. Honour [24]

As Dallosta and Simcik [20] pointed out, costs are committed early in a typical defense system project or program. Figure 12 identifies the cost justification for an SE approach as a method to prevent cost overruns. Later case studies in this review document that the SE contribution to cost containment is attributable to complete specification of requirements and interfaces so that those requirements can be traced through testing and V&V. In addition, several case studies illustrate that the complete requirements and interfaces developed through an MBSE approach are directly attributed to preventing defects and rework.

Honour reinforced his cost comparison and further justified employing an SE approach by correlating actual/planned schedule compliance against SE effort (Figure 13). As with the cost comparison, note two items of interest: 1) as SE effort increases, schedule compliance improves; and 2) there is a point when the schedule no longer improves with more SE effort. If we assume that schedule overruns also result in cost overruns, as is usually the case for defense systems, the chart of Dallosta and Simcik regarding cost commitment (Figure 6) adds additional relevance.

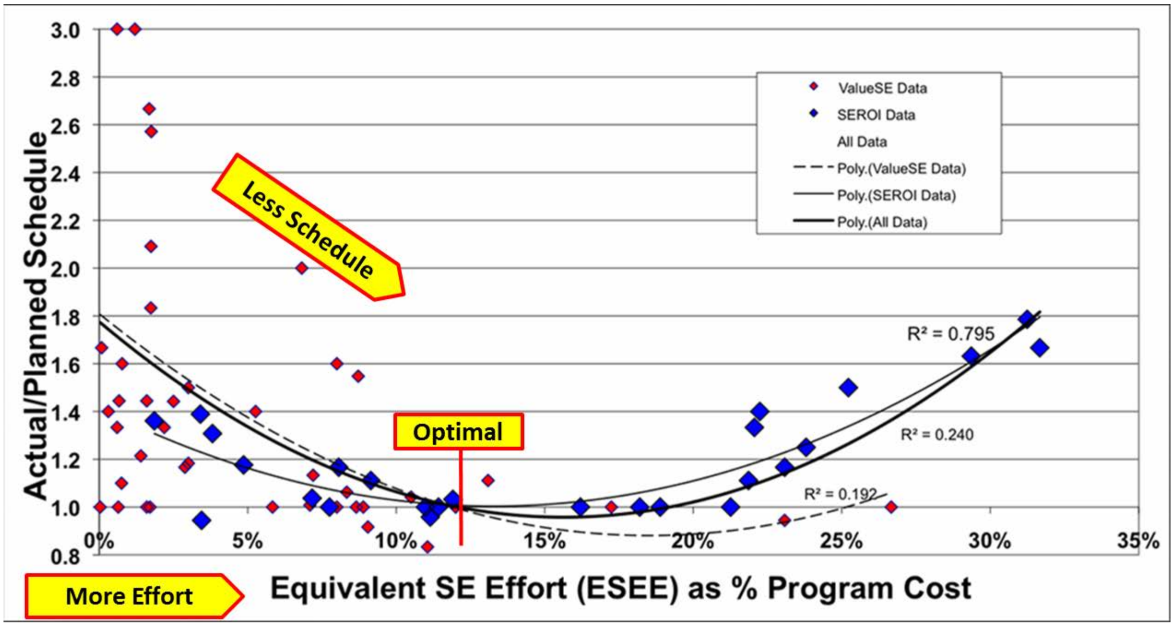

Figure 13: Systems Engineering Effort vs. Schedule
Adapted with permission from E. Honour [24]

Using the data from the projects plotted in Figure 12 and Figure 13 Honour tabulated ROI for SE efforts (Table 3).

> This calculation supports two strong findings. First, the monetary Return on Investment of greater systems engineering effort can be as high as 7:1 for programs using little to no current systems engineering effort. Second, the monetary Return on Investment of greater systems engineering effort for median programs is 3.5:1. [16]

Current SE Effort (% of Program Cost)	Average Cost Overrun	ROI for Additional SE Effort (Cost Reduction Per $$ Added)
0%	53%	7.0
5%	24%	4.6
7.2% (median of all programs)	15%	3.5
10%	7%	2.1
15%	3%	-0.3
20%	10%	-2.8

Table 3: ROI for Additional Systems Engineering Effort
Adapted with permission from E. Honour [16]

The thesis by Honour [16] is perhaps the most complete and well-documented justification for an SE approach found in this systematic literature review. Note that Honour correlated cost and schedule performance against any SE resource utilization. His thesis did not consider whether or not the project was using an MBSE approach. However, he documents not only a significant trend in cost and schedule improvements as SE effort is increased, but also the optimal point of SE effort as a percentage of total effort. This information, plus his documented 7X cost reduction per dollar for additional SE effort provides very compelling justification for an SE approach.

3.2.2.4. MBSE is an Improvement Over Systems Engineering

Tommasi and Vacca [25] justified an MBSE approach by documenting the level of improvement achievable from implementing SE, to MBSE, and to model-based product line engineering (MBPLE) approaches in their case study, "How Model-Based SE Makes Product/System Lifecycle Engineering Framework More Effective." They emphasized that an MBSE approach should be considered as an extension of SE, and that a MBPLE approach was a further extension of an MBSE approach.

Figure 14 shows comparable development cost between the three approaches on the left and comparable project on-time delivery on the right. For example, projects using an MBSE approach cost 55% less than projects using a traditional SE approach. In addition, projects using an MBSE approach delivered on-time 62% of the time, compared to 59% of the time with a traditional SE approach. Tommasi and Vacca drew data from an independent survey by Embedded Market Forecasters (EMF) of 667 SE respondents working on software-intensive product delivery projects.

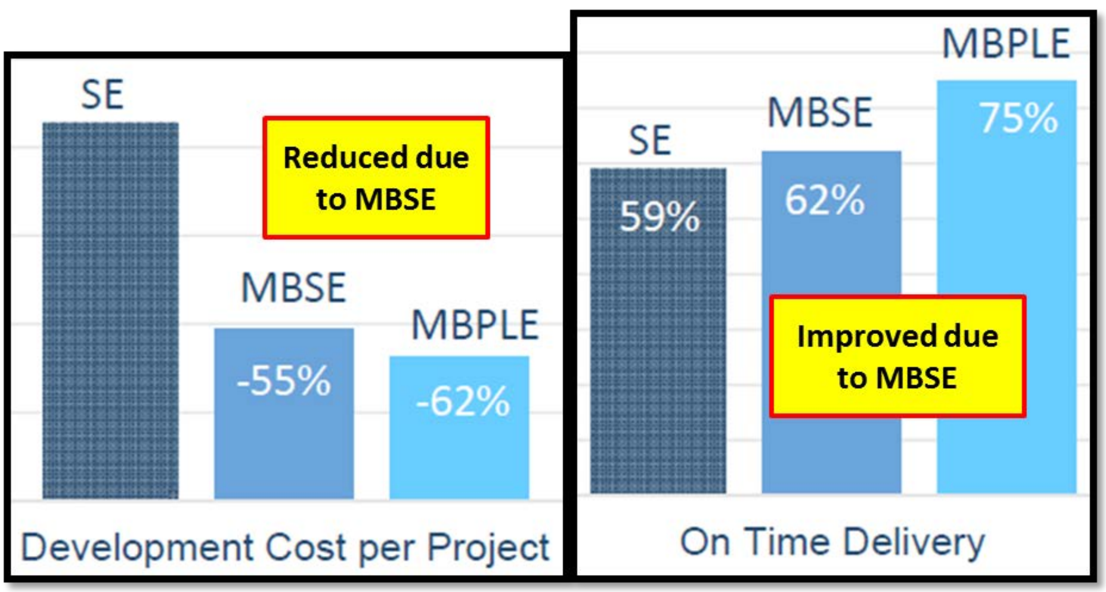

Figure 14: Effectiveness of Model-Based Systems Engineering
Adapted with permission, copyright © 2014 PTC, Inc. [25]

> "Extending a traditional PLE framework through adoption of Systems Engineering and Model-Based Systems Engineering methodologies multiplies its typical benefits." [25]

The authors asserted that as systems become more complex projects are increasingly challenged to provide a holistic view throughout the system life cycle in order to meet customer needs, reduce and mitigate risks, increase reuse, support design variants, and understand trade-offs. They asserted that an MBSE approach is critical to a successful MBPLE approach.

3.2.2.5. Modeling the Entire Engineering Process, and Halving the Effort

Sweetman [26] described perhaps the most advanced example and justification of an MBSE approach in his case study, "Economy Class: Saab plans to contain JAS 39E costs." Saab has implemented a fully integrated and automated implementation of an MBSE approach overlaying a 3-D model-based engineering (MBE) approach. The MBSE approach models the overall system requirements, architecture, tests, and V&V processes integrating all system components; while the MBE approach models the detailed component designs and automates the manufacturing processes. This full-system approach contributed directly to the Saab Aerospace Company's plans to significantly lower costs of development, procurement, and operation for the JAS 39E Gripen fighter plane.

> The most important tools are grouped under the term model-based systems engineering. Saab uses industry-standard Dassault Systemes Catia design software but says it applies it in unique ways. For example, there are no 2-D drawings in the JAS 39E program. Every part and manufacturing operation is defined by a 3-D model, from requirements and standards through design, manufacture and assembly and into the maintenance stage.
>
> The same model is used by all the groups involved in the design process – weight and balance, aerodynamics, weapon integration and so on. The result is that 70% of defects are discovered in the simulation stage and all groups can contribute to the solution and confirm that it will work. With earlier program tools, the design would be in flight-test by the time 70% of problems were identified.
>
> The definition of the Gripen C/D configuration includes 70,000 written documents. There are none in the JAS 39E database: Specifications and requirements (for example, resistance to bird-strike, corrosion and electromagnetic interference) are built into the models.
>
> The industry standard is to reach a near-optimal time for manufacture at the 180th aircraft produced. Saab wants to reach that stage by the 30th aircraft – halving the number of work hours taken on the first 100 aircraft.
>
> [26]

This case study is good example of the benefits of integrated development in a centralized modeling environment over a more traditional document centric approach. The authors of this literature review are aware of other similar examples of full system modeling in the aerospace industry but were unable to locate appropriate case studies in the time allotted. Further research may be warranted into how the aerospace industry is using an MBSE approach.

3.2.3. Justification Based on Reducing Defects and Preventing Rework

Eleven case studies justified either an SE or MBSE approach by emphasizing that these approaches enabled engineers to find defects early in the project development life cycle, where they are less expensive to fix and thus avoid costly later phase rework, which would impact both cost and schedule. Pena and Valerdi [27] correlated the impact from requirements volatility to the amount of SE effort. Boehm et al. [28] illustrated the impact of fixing defects early to delivery schedule and Hitchins [29] correlated the impact of fixing defects on overall project timelines. Chodas [30] illustrated how an MBSE approach can positively impact cost and schedule by limiting rework. Miller [31] illustrated how an MBSE approach can automate the tracking of defects. Tyreman et al. [32] documented the cost to fix defects. Maurandy et al. [33] compared attributes of a DBSE to those of an MBSE approach. Perez [34] applied an MBSE approach to risk-informed design. Ward and Redman [35] documented the cost avoidance potential of using an MBSE approach during the systems operation and support phase. Mitchell [36] documented the incremental improvement achieved after implementing an MBSE approach. And Saunders [22] compared defect density between DBSE and MBSE approaches.

3.2.3.1. Requirements Volatility Causes Defects and Increases Effort

Pena and Valerdi [27] justified a se approach in their study, "Characterizing the Impact of Requirements Volatility on Systems Engineering Effort." Figure 15 illustrates how volatility oscillates through the project life cycle for seven sample projects (each line represents a project).

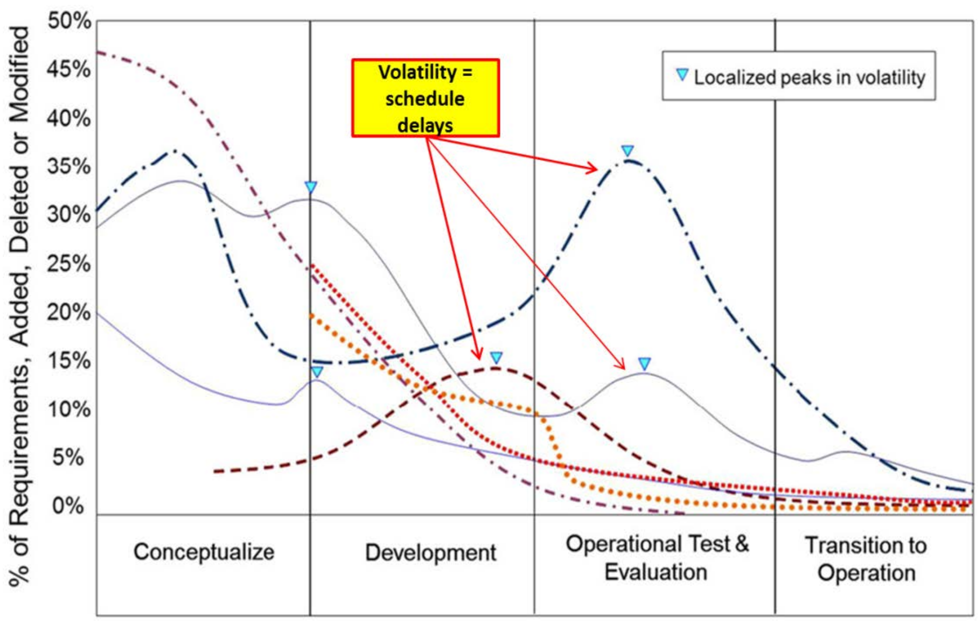

Figure 15: Requirements Volatility
© 2014 Wiley Periodicals, Inc. and Adapted with permission from M. Pena and R. Valerdi [27]

It was noted that some volatility early in the project is to be expected, but as the project timeline progresses volatility increasingly impacts project performance, due to an increase in SE effort in later life cycle phases to resolve omissions, defects, and rework. The y-axis in Figure 16 represents the average order of magnitude increase (e.g., 2 = 2x effort, 4 = 4x effort) in SE effort to resolve issues due to volatile requirements at each project phase.

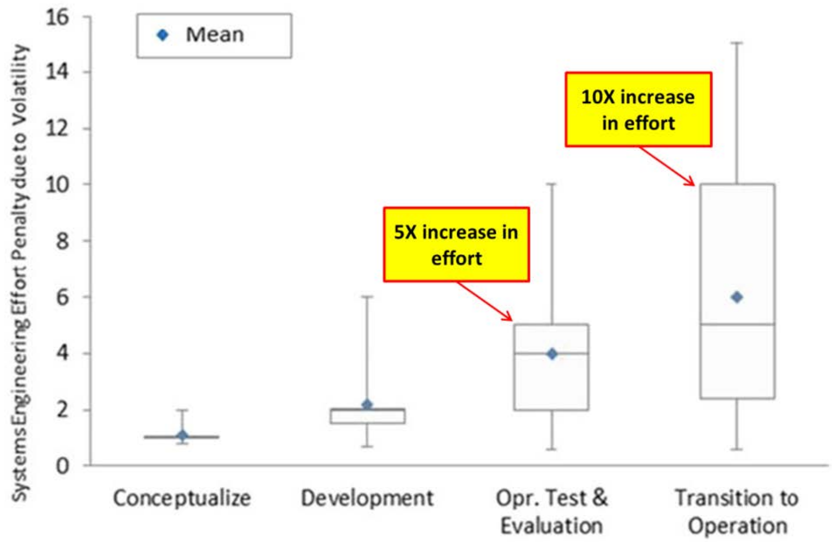

Figure 16: Impact of Volatility on Systems Engineering Effort
© 2014 Wiley Periodicals, Inc., and adapted with permission from M. Pena and R. Valerdi [27]

An SE effort is typically planned to trail off toward the later phases, as illustrated by Mornas et al. [17]. Pena and Valerdi illustrated that requirements volatility can cause a 10X increase in SE effort in the later phases (Transition to Operation). A 10X increase in SE effort late in the SDLC would likely cause a significant cost overrun.

3.2.3.2. Cost to Fix Defects Can be 100 Times More in Late SDLC Phases

Boehm et al. [28] justified an SE approach by correlating the relative cost to fix defects to project phase in their study, "The ROI of Systems Engineering: Some Quantitative Results for Software-Intensive Systems." Illustrated in Figure 17 is the increasing relative cost to fix defects as a system progresses through life cycle phases.

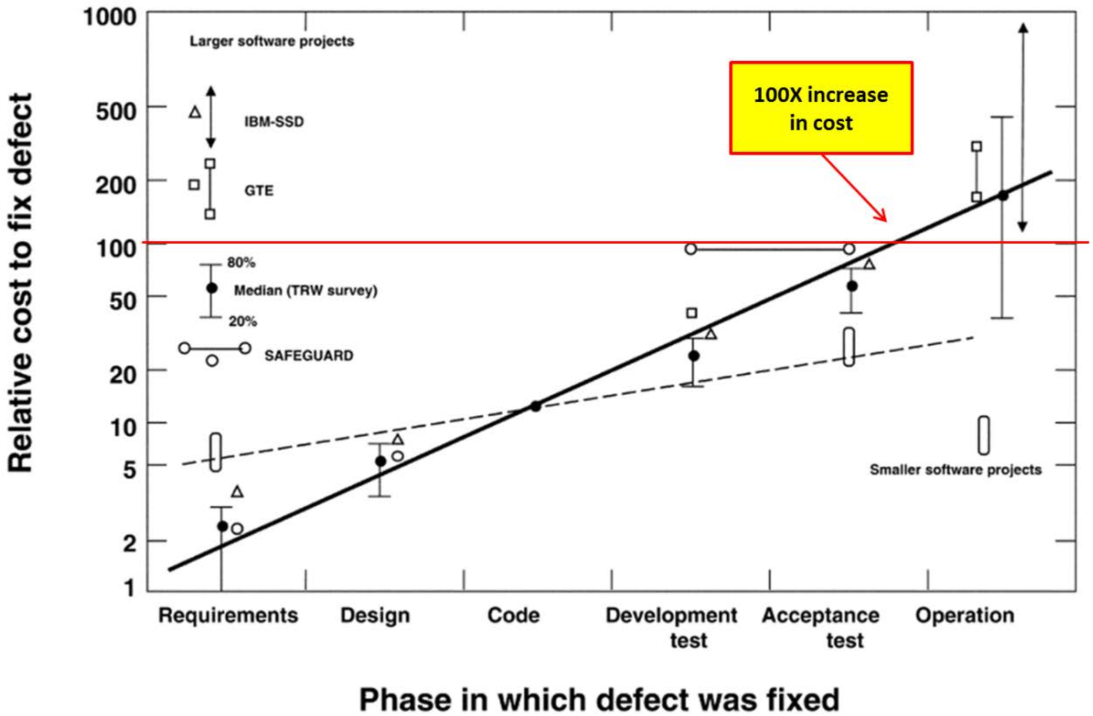

Figure 17: The Cost to Fix Defects by Phase
© 2008 Wiley Periodicals, Inc., and adapted with permission from* B. *Boehm et a. [28]

The specific amount of increase reported by different case studies reflects the differences in systems, complexity, labor (size and experience), technologies, and industries. However, the message is the same for each case study; that as the project progressed through the SDLC, the cost to fix defects increases by orders of magnitude.

Boehm et al. also justified an SE approach by illustrating the impact of defects on schedule in a project where emphasis was placed on finding and fixing defects early in the project life cycle. Note in Figure 18 the relatively low flattening of the ratio line – hours to fix: schedule – from 20 to 36 months through the development life cycle; less than the cost to fix defects during design. This indicates that the defects found during implementation were minor.

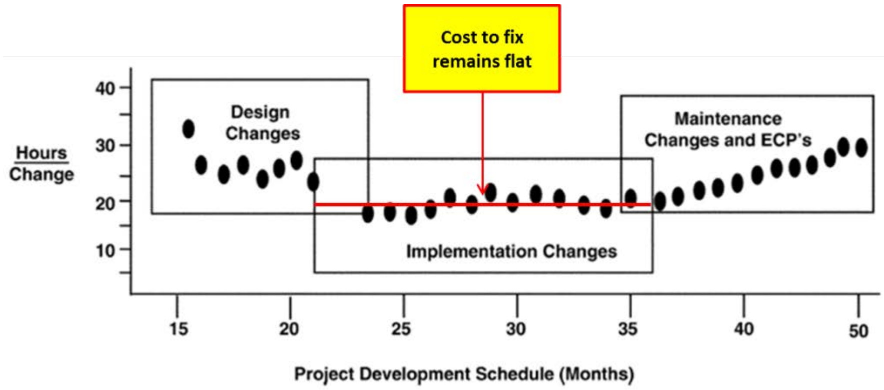

Figure 18: Flattened Cost When Defects are Fixed Early
© 2008 Wiley Periodicals, Inc., and adapted with permission from B. Boehm et al. [28]

Hitchins [29] justifies an SE approach by illustrating in his study, "Systems Engineering in Search of the Elusive Optimum," the impact to total project schedule when defects are found (and fixed) early. Figure 19 provides further evidence that the earlier a defect can be found and fixed the less impact the rework effort has on the project timeline.

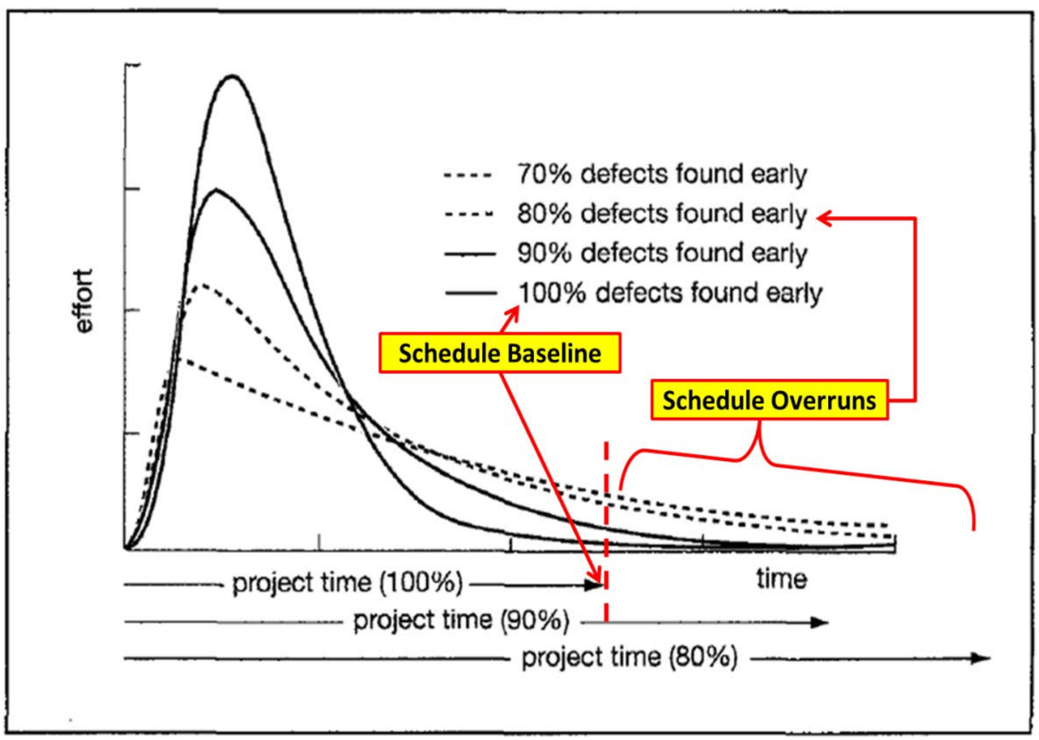

Figure 19: Rework Causes Project Overrun
© D K Hitchins [29], adapted with permission

In Figure 20, Hitchins further correlates the percentage of defects (initial errors) found early to the average amount of rework, and to the percentage of overrun, further illustrating that as the number of defects (errors) increase, so does the amount of rework and the corresponding percentage of project overrun.

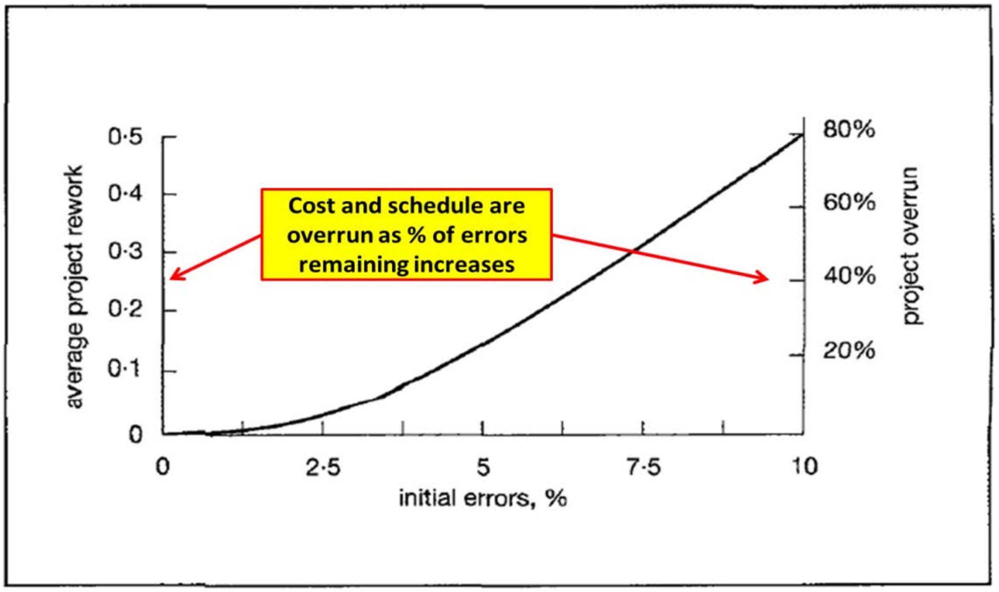

Figure 20: Escalating Cost Impact from Defects
© D K Hitchins [29], adapted with permission

Hitchins qualifies his findings with this statement:

> "…curtail the requirements phase …invariably leaves requirement and design specifications with errors and omissions" [29]

These case studies illustrate that finding and fixing defects early are important keys to maintaining project performance within costs, schedule, and quality objectives. The work of Boehm et al. [28] and Hitchins [29] sets the baseline for case studies, such as Chodas [30] (Section 2.2.2.3), which documents that an MBSE approach is effective at finding defects early and thereby enabling fixes early and/or preventing rework.

3.2.3.3. MBSE Improves Design Decisions

Chodas [30] justified an MBSE approach by illustrating improvements in cost and schedule in his case study, “Improving the Design Process of the REgolith Imaging X-ray Spectrometer (REXIS) Using Model-Based Systems Engineering (MBSE).” He set the stage by showing (Figure 21) that NASA had a significant number of projects with cost and schedule overruns.

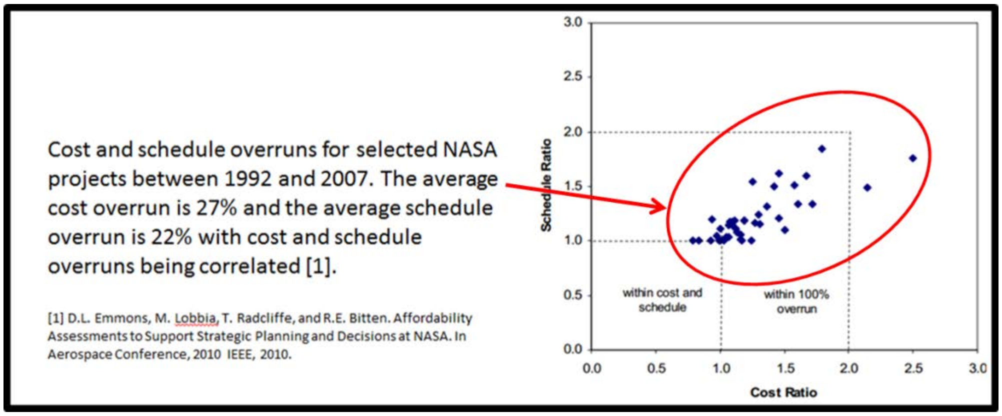

Figure 21: Cost Overruns at NASA
Published in the Public Domain [30]

Chodas explained that many systems at NASA are system-of-system projects and attributed the cost and schedule overruns to the growth of new subcomponents (parts) added to the system design in order to solve problems found as a result of design defects, omissions, or changes. He showed that an MBSE approach can have a significant positive impact on project cost and schedule by limiting the amount of rework as an improvement above using a traditional DBSE approach. Figure 22 illustrates how component (parts) growth often occurs after system design review (SDR), a point after which it becomes increasingly more costly to make changes.

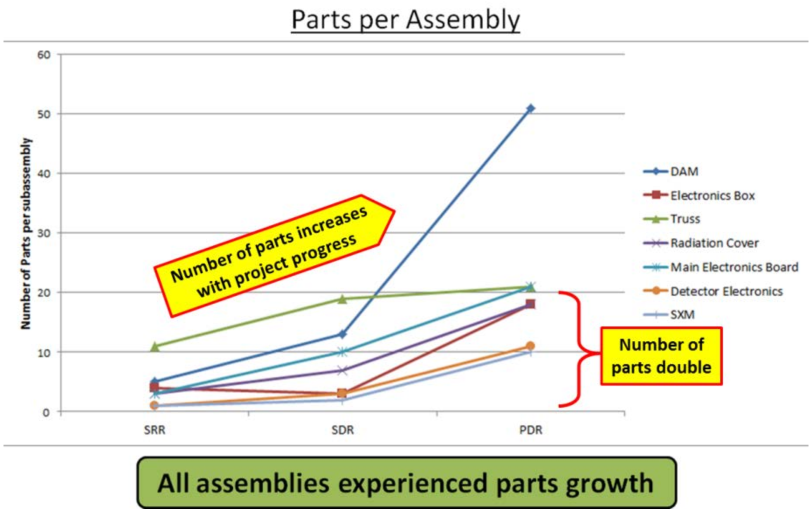

Figure 22: Part Growth After SDR
Published in the Public Domain [30]

In Figure 23, Chodas provides an example of how his team successfully used an MBSE approach to find a design solution sooner than they would have found using a DBSE approach. The MBSE approach found a requirement change that had not been reflected in the design (removal of spacecraft thermal isolation). The historical thermal design timeline is presented in the top half of the Figure 23. In the bottom half, the MBSE design timeline shows that the requirement change was recognized prior to SDR and the determination to change the design (adding a second isolation layer) is made in an earlier phase prior to the product design review (PDR). The result is reduced design iteration (which becomes increasingly more difficult after PDR approval) and rework prior to critical design review (CDR). Chodas does not identify what the cost saving were for making the design change earlier, but applying the lessons from Hitchins and Boehm et al., the cost savings can be presumed as substantial.

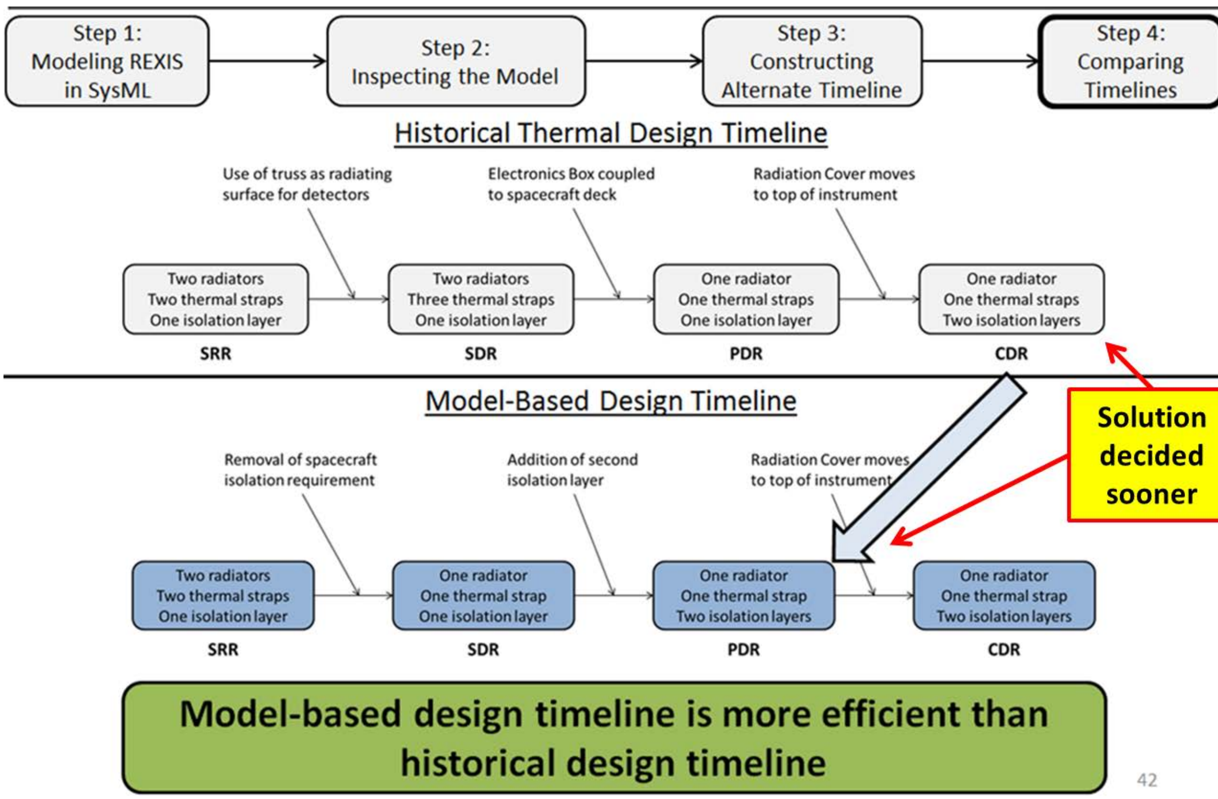

Figure 23: Example of MBSE Enabling Better Design Decisions
Published in the Public Domain [30]

Chodas' example illustrates how to justify an MBSE approach by walking through the process of preventing rework. His comparison with historical implementations of radiator thermal systems is relevant, because it illustrates the effect of a requirement omission in the historical view compared to requirements completeness in the MBSE view. Explicit requirements and interface tracing in an MBSE approach ensures that omissions do not happen, enabling design change decisions to happen earlier in the life cycle and preventing costly rework.

3.2.3.4. MBSE Modeling Software Automates the Finding and Tracking of Defects

Miller [31] justified an MBSE approach by presenting data from SE modeling tools used to automatically track defects to requirements, drawings, dimensions, parts, etc. in his case study, "How Has Effective Systems Engineering Benefited Our Defense Programs." Figure 24 illustrates an example of the data tracking possibilities from an MBSE tool, which provides the information necessary to find defects early in the systems development life cycle.

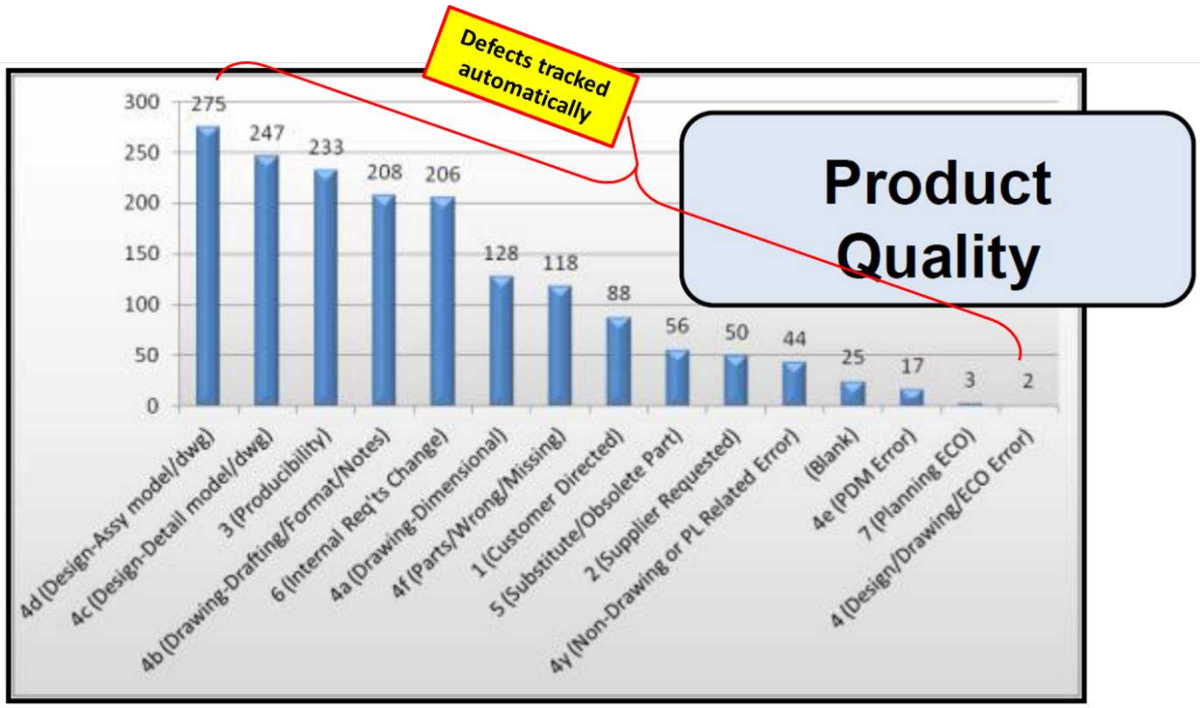

Figure 24: Automated Error Tracking Results Report
© Harris, Inc., 2012 [31]

Miller's case study builds on the work of Chodas [30] by illustrating data indicators that should be investigated (for example, the large number of defects in the Design-Assy and Design-Detail models) leading to the discovery of defects early in the SDLC.

3.2.3.5. MBSE Reduced the Cost to Fix Defects on a Complex Submarine Program

Tyreman et al. [32] justified an MBSE approach by illustrating in their case study, "Achieving MBSE Benefits amidst Multiple Government Program Office System-of-System Challenges," how an MBSE approach reduced the impact of fixing defects on a complex system-of-system submarine program. Program components included 4 ships, over 55 subsystems, and 25 original equipment manufacturers (OEMs). The group was chartered for a two-year technology renewal cycle and committed to doing more each cycle for the same price. The delta of engineering change proposals (ECPs) was the basic counted metric. The group solution was to meet this commitment by reducing their defect rate.

In Figure 25, the authors compare a traditional SE approach (sand colored section) against an MBSE approach (light green section). They illustrate how the traditional approach had much higher cost to fix defects than in the MBSE approach.

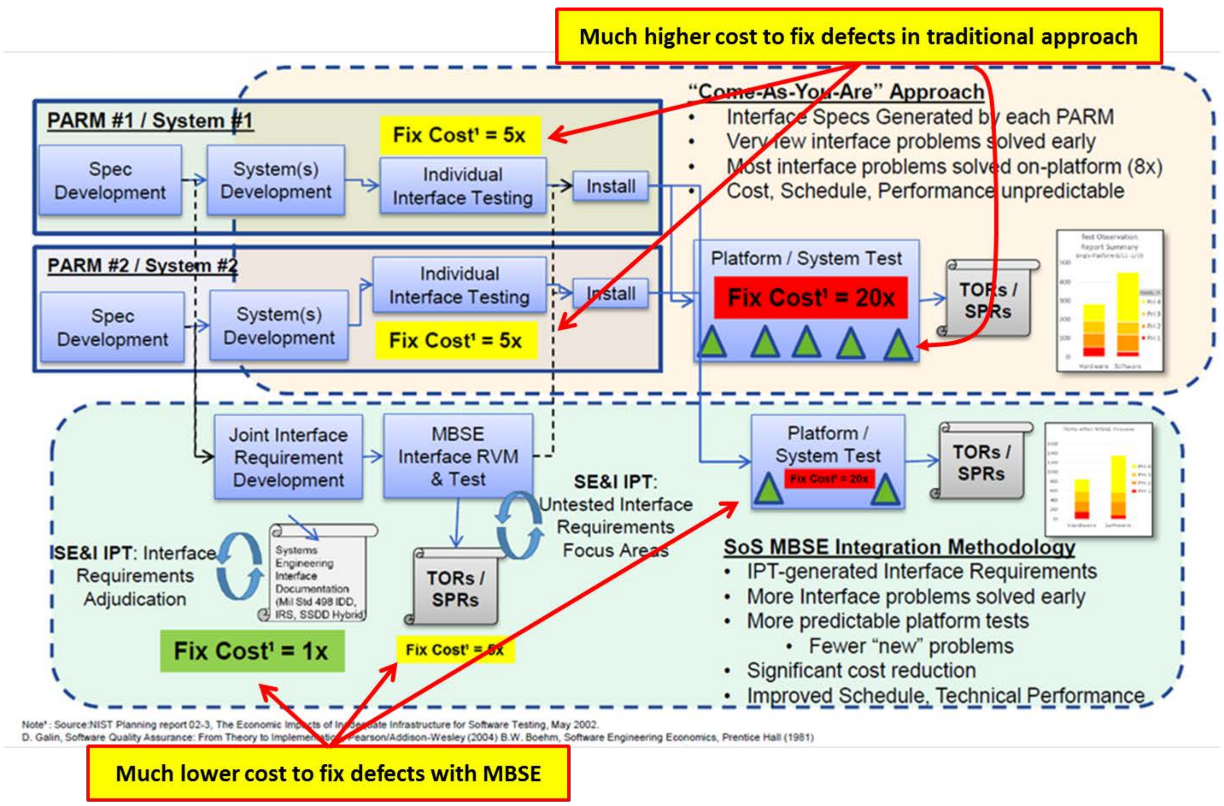

Figure 25: MBSE Finds Defects Earlier, Reducing Cost to Fix
Published in the Public Domain [32]

The impact estimates in Figure 25 (1x to 20x) were validated on the program, and further confirmed the results of Boehm et al. [28] (and others) that the earlier in the project timeline a defect is found, the less expensive it is to fix. Tyreman et al. justify MBSE as an approach to finding defects early. This case study is an example of an MBSE approach scaled to an extremely large and complex submarine system-of-system program where reducing the defect rate on this multi-billion dollar program would save many millions of dollars or more.

3.2.3.6. The Benefits of MBSE are Not Seen if Only Used for Requirements

Attempting to justify an MBSE approach, Maurandy et al. [33] conducted a cost-benefit analysis on using the systems modeling language (SysML) in their case study, "Cost-Benefit Analysis of SysML Modelling for the Atomic Clock Ensemble in Space (ACES) Simulator." ACES is an initiative of the European Space Agency to compare the accuracy of orbiting atomic clocks with ground-based clocks.

A basic assumption of this case study was that both DBSE and MBSE approaches are comparable in their ability to store information in order to communicate ideas to others. Attributes that reflect this concept for both approaches were defined.

The attributes measured are depicted in Table 4. The authors derived a weighting measurement for comparing the value of assigned attributes of requirements management in an MBSE approach to a DBSE approach. The comparison only considered the use of MBSE as an approach for requirements management and did not consider other SE functions.

	Weight
Completeness	0.097
Consistency	0.261
Extendability	0.048
Readability	0.161
Layering	0.433

Table 4: Weighted Requirements Attributes
© 2012 by Julien Maurandy, Ebehard Gill, Achim Helm, and Roland Stalford [33]

> "The results are $R_{MBSE} = 0.489$ and $R_{DBSE} = 0.511$. Looking closer, one can quickly realize these coefficients highly depend on the ratio of the total project cost to the number of team members."
>
> "Architecture and Behavior Context: $F_{MBSE} = 1.15$ and $F_{DBSE} = 0.85$
> Requirements Handling Context: $F_{MBSE} \approx 1.02$ and $F_{DBSE} \approx 0.98$" [33]

The authors readily admit that their results showed marginal differences between MBSE and DBSE approaches. This case study was the only case study found in this literature review that documented a less than obvious improvement from using an MBSE approach. Therefore, it provides a counter argument to the inherent biases of the other case studies.

However, this case study further illustrates that using an MBSE approach only for requirements management may achieve only marginal benefits. Advantages from an MBSE approach such as ensuring completeness of requirements, interfaces, and design elements, thus enabling omissions, inconsistencies, and defects to be found early in the life cycle cannot be leveraged into later phases where the cost and schedule savings or rework prevention is realized. Similarly, this case study ignored the advantages from reusing models. Therefore, a program that does not continue an MBSE approach throughout the full system development life cycle will likely incur all of the investment cost, but will not reap the benefits or ROI hoped for.

3.2.3.7. MBSE Improves the Probability of Success

Perez [34] justified an MBSE approach in his case study, "Application of MBSE to Risk-Informed Design Methods for Space Mission Applications," by documenting how MBSE tools and processes improve the probability of success by reducing risk in design decisions. Risk-Informed Design (RID) is an analysis method employed early in the lifecycle of spaceflight projects, enabling designers to include a risk factor in their component trade-off decisions.

Perez explains that risks analyses are typically conducted late in the design cycle, when design changes are difficult to implement. To improve this process the team evaluated the use of failure analysis integrated into the system architecture model and the use of an MBSE approach to enable engineers to make risk-informed system modification decisions during the design process.

Table 5 summarizes the risk probabilities for adding alternative subsystems to a larger system. The baseline probability is for no change to the system design. Update 1 reflects the probability for system changes derived using an SE approach, without models (improving the likelihood of success to 73%), and update 2 reflects the probability for system changes derived while using an MBSE approach (improving the likelihood of success to a 93%).

	System f0 Phase	Success Probability [probOfSuccess]	Failure Probability [1 - probOfSuccess]
No change to design	Baseline	0.6	0.4
Without MBSE	Update 1	0.73	0.27
With MBSE	Update 2	0.93	0.07

From 73 % chance of success to 93 % chance of success

Table 5: Probabilities from Implementing MBSE
© 2014 by Rafael Mareni Perez [34]

However, the authors provided this caveat about adding subsystems to a space system design:

> Although the reliability of the system improved significantly, it came at the sacrifice of additional complexity to the system that typically results in higher costs, longer schedules, higher mass, larger volume, more electrical power and/or extra resources. [34]

The use of an MBSE approach did not reduce cost directly; rather the use of an MBSE approach allowed the authors to make more informed design option decisions. The more informed decision significantly improved mission reliability (reducing risk) and prevented a cost increase later due to potential rework or due to a system failure, which if it occurred in space would likely have been catastrophic.

3.2.3.8. An ROI on the Cost to Develop a Custom MBSE Approach

Ward and Redman [35] documented the ROI to develop and implement the System Architecture Virtual Integration (SAVI) MBSE approach (tools, standards, labor, etc.) by calculating the potential rework cost avoidance that using SAVI could achieve. The program is

> A collaboration between aerospace system development stakeholders whose goal is to lower development costs of complex aerospace systems by enabling model-driven virtual integration of complex systems across multiple development environments. The SAVI Program resides within the Aerospace Vehicle Systems Institute (AVSI), an aerospace industry research cooperative whose members perform collaborative applied research and technology projects. AVSI is part of the Texas A&M Engineering Experiment Station. Current Members of the SAVI Program include Airbus, Boeing, U.S. DoD, Embraer, U.S. FAA, GE Aviation, Honeywell, U.S. NASA, Rockwell Collins, the Software Engineering Institute at Carnegie Mellon University and United Technologies Corporation. [www.savi.avsi.aero]

The SAVI program was initiated in response to the growing industry trends toward the use of models for SE, development, manufacturing, production, verification and validation (V&V), and integration. The authors argue that as models proliferate, the various models contain multiple interdependent properties, resulting in multiple versions of the truth. In order to maintain a single version of truth the complete model set for a system needs to be consistent. SAVI is an MBSE approach aimed at providing consistency across all models.

In Table 6, the authors illustrate their results from the use of parametric estimation software, where they calculated a risk factor of the cost of rework and amortized that cost over 10 years. Since these are multi-year calculations, they have equalized the values to net present value (NPV). The cost avoidance estimates are used to calculate an ROI, which for the SAVI approach shows a positive ROI to implement this MBSE approach even in the most pessimistic estimates.

	NPV (Cost Avoidance)	Total Cost Avoidance	NPV (Cost to Develop)	ROI % per year
Pessimistic	$64 M	$99 M	($85.7 M)	2%
Expected	$256 M	$398 M	($85.7 M)	40%
Optimistic	$768 M	$1.193 B	($85.7 M)	144%

ROI to develop the SAVI MBSE approach

Table 6: ROI to Develop a Custom MBSE Approach
Adapted with permission from the SAVI PMC [35]

This case study provides representative cost ($85.7M) and ROI (40%) for developing and implementing a complex MBSE approach for a large aerospace systems development operation.

3.2.3.9. MBSE Allows a Reduction in SE Labor, Even as BCRs Increase

Mitchell [36] justified their transition from a traditional DBSE approach to an MBSE approach by documenting the incremental improvement that his team experienced during their transition on the Submarine Warfare Federated Tactical Systems (SWFTS) Program. He describes using continuous process improvement techniques on their SE processes in response to request from their customer to do more for less cost. As is normal in a large complex weapon system development program, the number of baseline change requests (BCR) increased steadily throughout the course of the multi-year program. Doing more for less did not prevent the SWFTS program from investing in new processes and tools. Notice the bump in Figure 26 (blue line) after implementing the DOORS tool. Mitchell explains that extra effort was required to implement the new processes and tools. Lessons learned from that effort helped smooth the transition to an MBSE approach in 2011. Mitchell points out that through improved modeling and automation of the SE processes, his team was able to reduce the number of SE full-time equivalents (FTE) assigned to the project, even though the number of BCRs continued to increase (doing more for less cost).

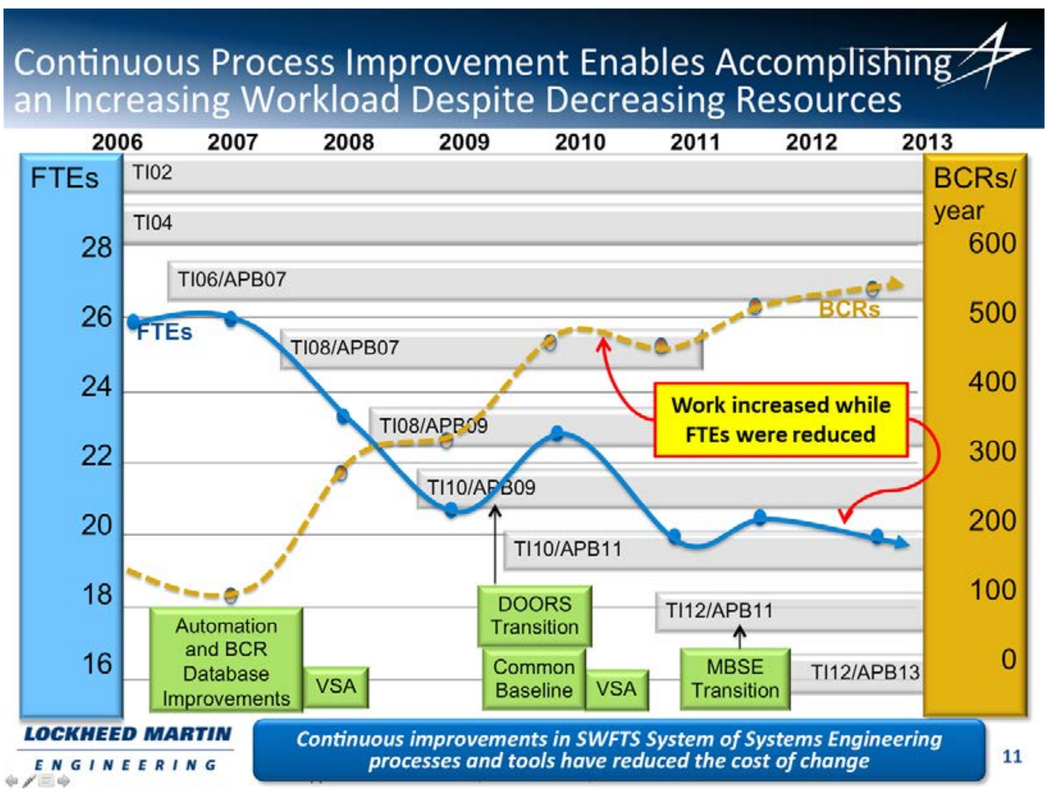

Figure 26: MBSE Enabled a Reduced Systems Engineering Effort
Published in the public domain and adapted with permission from S. Mitchell [38]

While controlled experiments providing direct comparisons of large side-by-side development projects are not financially feasible, the types of results documented by Mitchell are evidence of the value of an MBSE approach, where the team was able to do more work with fewer members.

3.2.3.10. A Dramatic Reduction in Defects Directly Attributed to MBSE

Saunders [22] justified an MBSE approach in his case study, “Does a Model Based Systems Engineering Approach Provide Real Program Savings? – Lessons Learnt,” by comparing defect rates on four programs run under a DBSE approach to three programs using an MBSE approach. Figure 27 illustrates his dramatic 68% reduction in defects after MBSE practices were introduced. Defects were tracked over a five-year period, starting before an MBSE approach was implemented.

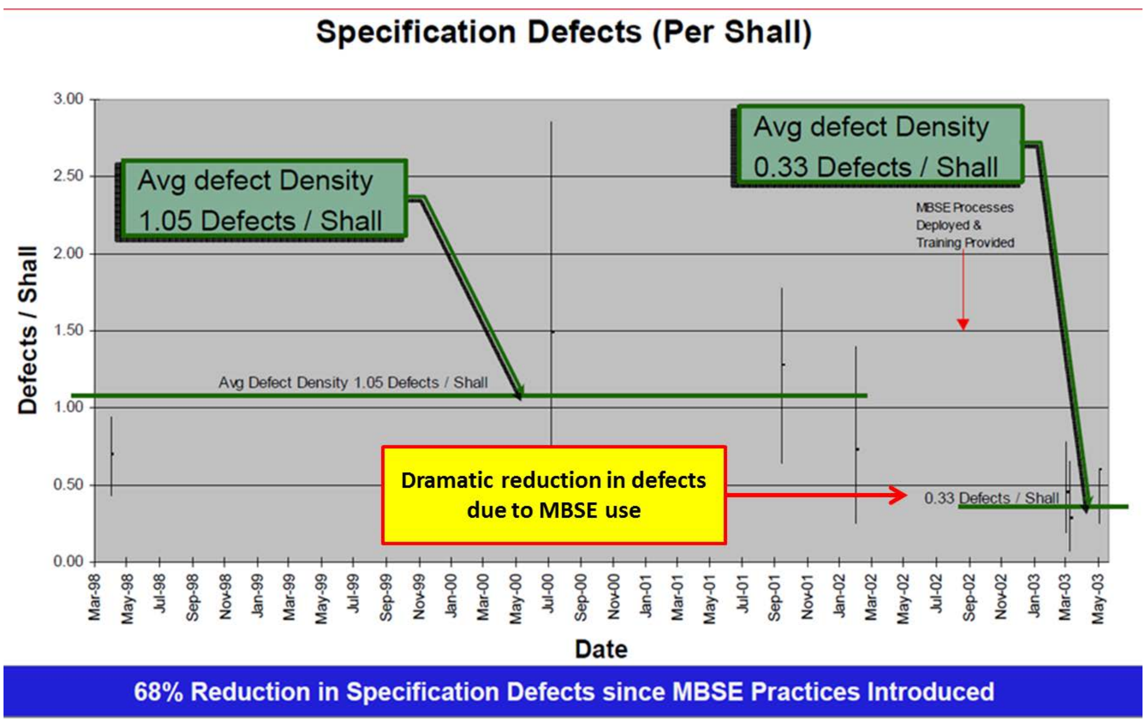

Figure 27: Reduced Defects After Introducing MBSE
© 2011 Raytheon Company and adapted with permission from S. Saunders [22]

Saunders describes how the DBSE approach is focused on producing the specification, perhaps using a requirements management tool (which he does not consider an MBSE tool). He further explains that the largely manual processes involved in a DBSE approach produce poor traceability reports and incorrect specifications. Saunders contrasts the MBSE approach as providing complete understanding of the system behavior. “An MBSE approach,” he states, “focuses on evolving the system model until all views are completely integrated” (functional, requirements, architecture, and test). He emphasizes, “functional behavior cannot be expected to be understood to the extent needed to create a complete and consistent specification,” (with a DBSE approach).

As with Mitchell’s [36] case study, Saunders attributed improvements in the reduction of defects due to the use of an MBSE approach. Saunders’ comparison of only seven sample projects is dramatic. It would be worthwhile to conduct further research. Perhaps a follow-on study could show whether the improvements continued into 2015.

4. Conclusions

The genesis for this systematic literature review was to search for industry case studies that could inform a decision of whether or not to support the change process, investment, training, and tools needed to implement an MBSE approach across the engineering enterprise.

There is a significant advantage to project performance by applying an SE/MBSE approach. A ratio of 12:21 case studies reported that applying an SE approach established repeatable processes with strong project performance results for requirements management, architecture, testing, and V&V processes. A ratio of 11:21 case studies reported that applying an MBSE approach significantly improved project performance above that of an SE approach. This was attributed to added processes, for stakeholder and requirements stability, concept exploration, design reuse, and margin analyses. These are the processes where an MBSE approach enables defects to be found and fixed early in the SDLC. Defects and rework are the cause of significant project performance degradation, and an MBSE approach was shown to prevent this impact.

An MBSE approach improves engineering efficiency. The case studies that did not include quantifiable metrics primarily stated that an MBSE approach could make a complex system development effort more efficient. The most commonly discussed application of an MBSE approach in these case studies was for requirements completeness, consistency, and communication. These were seen to improve engineering processes involved in requirements management, test and evaluation, V&V, concept exploration, design reuse, system margins analyses. Improvements in defect prevention strategies were made by identifying, analyzing, and tracking defects, so that they are fixed in the phase where the defects occur (where the fix will have the least impact to cost and schedule).

An MBSE approach prevents costly rework. The case studies with metrics, in a ratio of 12:21, presented evidence confirming that an MBSE approach provides significant capability to find defects early in the SDLC when they can be fixed with less impact, thus mitigating the risks to cost, schedule, and mission by preventing rework late in the life cycle. Rework late in the SDLC poses significant risks to the program for cost and schedule overruns, because many complex system programs commit total program cost in the early phases of the SDLC, leaving little room to absorb rework in the later phases.

Using an MBSE approach only for requirements management may achieve only marginal benefits. Advantages such as enabling defects to be found early in the life cycle cannot be leveraged into later phases where the cost and schedule savings or rework prevention are realized if a program only employs an MBSE approach for requirements management. A program that does not continue an MBSE approach throughout the full system development life cycle will likely incur all of the investment cost, but will not reap the benefits or ROI hoped for.

Systems engineers have a role in driving the engineering processes. Along with applying an SE or MBSE approach across the SDLC, the greatest performance success was achieved when the SE held a process leadership role. Mornas et al. [17] defined those processes that SEs should lead, which have effect across the SDLC. Research by Elm and Goldenson [18] showed that programs employing an SE approach only for documenting requirements rated in the low performance/low SE capability category, while those with a strong SE capability achieved significantly higher performance. Frantz [23] attributed success in the UHF project that delivered 3X sooner than the worst UHF project to high access and attention from systems management disciplines. The defined optimal SE effort (from 12%-17%) by Honour [16 and 24] assumed that the SE functions within this role as the driver of engineering processes.

There is a need for skilled engineers. A number of the case studies addressed the general lack of skilled systems engineers, and skilled MBSE engineers in particular, as a major hindrance to implementing an MBSE approach successfully. If the models are to provide benefit to later phases of the program, they must be managed and kept up to date. Design engineers need a basic understanding of an MBSE approach in order to fully utilize the information generated by systems engineers. Both engineers and stakeholders are accustomed to reviewing documents rather than MBSE model artifacts and the fundamental process of “first change the model, the model is the design” meets with cultural resistance. The greatest challenge to a successful implementation of an MBSE approach may be overcoming this cultural resistance.

Additional research is needed. Much of the analyses conducted in these case studies are incomplete when applied to an MBSE approach. For example, Honour [24] documented an optimal percentage of systems engineering effort correlated to project performance (from 12%-17%). An extension of his analysis specific to an MBSE approach would be informative, because of the process, schedule, and cost differences between the two approaches. Some of the defect analyses in other case studies used only a few samples in their report. It would be worthwhile to expand the sample size and see if the improved performance continues. Some of the defect analyses used samples from several years ago. It would be worthwhile to explore whether an MBSE approach continues to provide significant benefits over a longer period of time. While the findings of this literature review are positive, further study is warranted in order to establish a definitive ROI for implementation of an MBSE approach in the target environment, as well as account for differences in cultures, motivations, management styles, marketplace pressures, governmental regulations, and staff experience between the case studies reviewed here. What works for one company may not work for another or fit another’s business model.

5. Implementation Lessons Drawn from the Findings

The results of this study can be used to construct a picture of what it would take for an organization to successfully employ an MBSE approach and to achieve the benefits illustrated.

First, the case studies identified are a number of prerequisites for any enterprise to employ an MBSE approach, representing investments by the enterprise in its staff and processes.

- The enterprise must have a mature, well-documented, and enterprise-wide SE process that spans from requirements development and analysis through system test and qualification (or V&V in the case of software intensive systems).
- The enterprise must have a cadre of trained systems engineers with at least moderate skill in employing MBSE tools and techniques, and whose MBSE roles are clearly delineated from the more traditional roles.
- The enterprise must make available to all the engineering staff a basic level of training in the MBSE processes (so that they understand the value of the models and what to expect from the systems engineers) and in how to read MBSE artifacts (so that they can interpret information provided from the MBSE processes).
- The enterprise must define MBSE model management processes in order to create, update, and maintain the MBSE models throughout the full lifecycle and to derive engineering artifacts from the models at each stage of the system development lifecycle.
- The enterprise must invest in full-scale MBSE tools (not requirements management tools) and tool-use procedures must be institutionalized (all teams must use compatible tools).

In addition to these prerequisites, the case studies assert that each program in the enterprise would need to make the commitment to employ an MBSE approach in the following manner:

- Initiate MBSE model development at the beginning of the program to model the program processes (as well as requirements and other artifacts) and resource the SE effort sufficiently, including MBSE work (which could be as much as 15% of program costs).
- Commit to making the MBSE models the centerpiece artifact for referencing the architecture, converting requirements into designs, and tracing tests and V&V tasks through design to requirements ("first change the model, the model is the design").
- Provide continuous resources to update and sustain the models and employ them as the basis for design reviews and verifying requirements, using the models to improve understanding of the impact to the design caused by changes.
- Provide continuous resources to employ the models as the basis for defining system testing, qualification, and requirements verification plans.
- Support an MBSE approach with appropriate sustained computing infrastructure (licenses, repositories, access controls, archives, and processes) throughout the system life cycle.

Based on the case studies reviewed in this document, the enterprises which acquired significant benefits from an SE approach in general and an MBSE approach in particular made these investments and commitments. The cultural changes necessary to implement an MBSE approach successfully (roles, rewards, behavior, and support at all levels) were described as the more difficult challenges to overcome. However, those who remained committed to the application of an MBSE approach throughout the SDLC achieved significant ROI.

References

[1] Edited by D.D. Walden, G.J. Roedler, K.J. Forsberg, R.D. Hamelin, T.M. Shortell, *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, Fourth Edition. John Wiley & Sons, Inc., 2015.

[2] S.A. Sheard and C.L. Miller, "The Shangri-La of ROI." *Proceedings of the Tenth International Symposium of the International Council on Systems Engineering*, held in Brighton, England, June 2000. Software Productivity Consortium NFP, Inc., 2000.

[3] R. Reil, "Improved Traceability of Mission Concept to Requirements Using Model Based Systems Engineering," NASA technical report ARC-E-DAA-TN14160, April 3, 2014.

[4] K. Vipavetz, D. Murphy, and S. Infeld, "Model-Based Systems Engineering Pilot Program at NASA Langley," presented at the *AIAA SPACE 2012 Conference & Exposition*, held in Pasadena, California, September 11-13, 2012.

[5] M. Ryan, S. Cook, W. Scott, "Application of MBSE to Requirements Engineering—Research Challenges," presented at *2013 Systems Engineering and Test and Evaluation Conference* (SETE2013), held in Canberra, Australia. April 29-May 1, 2013,

[6] Y. Bijan, H. Graves, J. Yu, J. Stracener, T. Woods, "Using MBSE with SysML Parametrics to Perform Requirements Analysis," presented at the *2011 INCOSE International Symposium*, held in Denver, CO, June 20-23, 2011.

[7] E.A. Bjorkman, S. Sarkani, and T.A. Mazzuchi, "Using Model-Based Systems Engineering as a Framework for Improving Test and Evaluation Activities," *Systems Engineering*, vol. 16, November 3, 2013.

[8] D.R. Wibben and R. Furfaro, "Model-Based Systems Engineering approach for the development of the science processing and operations center of the NASA OSIRIS-Rex asteroid sample return mission," *Acta Astronautica*, Vol. 115, pp. 147-159, 2015.

[9] M.O. Khan, G.F. Dubos, J. Tirona, S. Standley, "Model-Based Verification and Validation of the SMAP Uplink Processes," *IEEE 2013 Aerospace Conference*, held in Big Sky, MT, March 2-9, 2013.

[10] T. Bayer, S. Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I. Gontijo, K. Lewis, M. Moshir, R. Rasmussen, and D. Wagner, "Early Formulation Model-Centric Engineering on NASA's Europa Mission Concept Study," presented at the *2012 INCOSE International Symposium*, held in Rome, Italy, July 9-12, 2012.

[11] H.G.C. Góngora, M. Ferrogalini, C. Moreau, "How to Boost Product Line Engineering with MBSE: A Case Study of a Rolling Stock Product Line," presented at the *5th International Conference on Complex Systems Design & Management*, November 12-14, 2014.

[12] T.J. Bayer, S. Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I. Gontijo, K. Lewis, M. Moshir, R. Rasmussen, D. Wagner, "Model Based Systems Engineering on the Europa Mission Concept Study," presented at the *IEEE 2012 Aerospace Conference*, held in Big Sky, MT, March 3-10, 2012.

[13] H.G.C. Góngora, A. Dauron, T. Gaudré, "A Commonsense-Driven Architecture Framework, Part 1: A Car Manufacturer's (naïve) Take on MBSE," presented at the *2012 INCOSE International Symposium*, held in Rome, Italy, July 9-12, 2012.

[14] K.G. Young, "Defense Space Application of Model-Based Systems Engineering (MBSE) – Closing the Culture Chasms," presented at the *AIAA SPACE 2015 Conference and Exposition*, held in Pasadena, California, August 31-September 2, 2015.

[15] K.A. Simpson, O.V. Sindiy, and T.I. McVittie, "Orion Flight Test 1 Architecture – Observed Benefits of a Model Based Engineering Approach," presented at the *IEEE 2012 Aerospace Conference*, held in Big Sky, MT, March 3-10, 2012.

[16] E.C. Honour, "Systems engineering return on investment," Ph'D thesis, Defence and Systems Institute, School of Electrical and Information Engineering, University of South Australia, 2013.

[17] O. Mornas, C.L. Weywada, R. Mazzella, A. Sigogne, P. Pancher, "Development of Systems Engineering People to Support Major Transformation Plans in Thales (Process, Roles, Methodology & Related Tools)." Article published and used by INCOSE with permission, [available online] at www.incose.org. 2012.

[18] J.P. Elm, D.R. Goldenson, "The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey." Special Report *CMU/SEI-2012-SR-009 CERT Program*. Carnegie Mellon University. November 2012.

[19] M. Bone, R. Cloutier, "The Current State of Model Based Systems Engineering: Results from the OMG™ SysML Request for Information 2009." *8th Conference on Systems Engineering Research*, held in Hoboken, NJ, March 17-19, 2010.

[20] P. Dallosta, T. Simcik, "Driving Reliability, Availability, and Maintainability in While Driving Cost Out," *in Defense AT&L*, Special Issue – Product Support, pp. 35-38, 2012. This article is in the public domain, per notice from *Defense AT&L*, © 2012.

[21] E. Tonnellier, O. Terrien, “Rework: models and metrics, An Experience Report at Thales Airborne Systems.” Published and used by INCOSE with permission [available online] at www.incose.org, pp. 1619-1632. 2012, © 2012 by Edmond Tonnellier and Olivier Terrien.

[22] S. Saunders, “Does a Model Based Systems Engineering Approach Provide Real Program Savings? – Lessons Learnt.” *Informal Symposium on Model-Based Systems Engineering DSTO*, held in Edinburgh, South Australia, October 25, 2011, © 2011 Raytheon Company, published in the public domain by *Defense AT&L*, © 2011.

[23] W.F. Frantz, “The Impact of Systems Engineering on Quality and Schedule *Empirical Evidence*.” Published and used by INCOSE, pp. 618-624 1995, with permission from The Boeing Company, © 1995, [available online at www.incose.org].

[24] E.C. Honour, Systems engineering return on investment (SE-ROI), Results from 15 years of research, with data from 94 programs,” [accessed online December 2015] www.hcode.com/seroi/.

[25] C. Tommasi, E. Vacca, “How Model-Based SE Makes Product/System Lifecycle Management Framework More Effective,” © 2014 PTC Inc., Professional Paper, November 24, 2014.
INFORMATION PROVIDED BY PTC IS FURNISHED FOR INFORMATIONAL USE ONLY AND PTC ASSUMES NO RESPONSIBILITY OR LIABILITY FOR ANY ERRORS OR INACCURACIES THAT MAY APPEAR IN THE DOCUMENT. THE MATERIAL FROM PTC COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS

[26] W. Sweetman, “Economy Class: Saab plans to contain JAS 39E costs.” *Aviation Week & Space Technology*. Vol. 176 Issue 11, pp. 48-49, March 31, 2014.

[27] M. Pena, R. Valerdi, Characterizing the Impact of Requirements Volatility on Systems Engineering Effort. Published online in Wiley Online Library (www.wileyonlinelibrary.com). August 19, 2014. DOI 10.1111/sys.21288, © 2014 Wiley Periodicals, Inc.

[28] B. Boehm, R. Valerdi, E. Honour, *The ROI of systems engineering: Some quantitative results for software-intensive systems,* © 2008 Wiley Periodicals, Inc., published online in Wiley *InterScience* (www.interscience.wiley.com), April 1, 2008, DOI 10.1002/sys.20096.

[29] D. Hitchins, “Systems engineering in search of the elusive optimum.” *Engineering Management Journal,* Vol. 8, Issue 4, pp. 195-207, August 1998, DOI: 10.1049/em:19980412.

[30] M. Chodas, “Improving the Design Process of the REgolith Imaging X-ray Spectrometer (REXIS) Using Model-Based Systems Engineering (MBSE).” *NASA GSFC Systems Engineering Seminar*, September 15, 2014.

[31] C. Miller, “How Has Effective Systems Engineering Benefited Our Defense Programs.” *NDIA SE Conference: Effective Systems Engineering,* © Harris, Inc. 2012.

[32] J. Tyreman, G. Saroch, R. Byers, Achieving MBSE Benefits amidst Multiple Government Program Office System of System Challenges. *NDIA - 18th Annual Systems Engineering Conference*, held in Springfield, VA, October 26-29, 2015.

[33] J. Maurandy, E. Gill, A. Helm, R. Stalford, “Cost-Benefit Analysis of SysML Modelling for the Atomic Clock Ensemble in Space (ACES) Simulator,” published and used by INCOSE with permission [available online] at www.incose.org, pp. 1726-1745, © 2012 by Julien Maurandy, Ebehard Gill, Achim Helm and Roland Stalford.

[34] R.M. Perez, “Application of MBSE to Risk-Informed Design Methods for Space Mission Applications.” *AIAA SPACE 2014 Conference and Exposition*, held in San Diego, CA, August 4-7, 2014, © 2014 by Rafael Mareni Perez.

[35] D. Ward, D. Redman “AVSI’s System Architecture Virtual Integration Program: Proof of Concept Demonstrations.” *Presentation to the INCOSE MBSE Workshop*, held in Jacksonville, FL. January 27, 2013. Additional program information available online at [savi.avsi.aero], AVSI © 2013. Permission granted by the SAVI Program Management Council (PMC), AVSI.

[36] S.W. Mitchell, “Transitioning the SWFTS Program Combat System Product Family from Traditional Document-Centric to Model-Based Systems Engineering, Journal of Systems Engineering,” Vol. 17, No. 3, Fall 2014, © 2013 Wiley Periodicals, Inc. Published online in Wiley Online Library (www.wileyonlinelibrary.com). June 18, 2013. DOI 10.1002/sys.21271.

[37] Edited by C. Haskins, K.J. Forsberg, M. Krueger, D.D. Walden, R.D. Hamelin, *INCOSE Systems Engineering Handbook: A guide for system life cycle processes and activities, Third Edition*, v.3.2.2. INCOSE-TP-2003-002-03.2.2, October 2011.

[38] S.W. Mitchell, “Transitioning a Combat System Product Family from Traditional Document-Centric Systems Engineering to MBSE,” published in the public domain at the *NASA JPL Seminar on Model-Based Systems Engineering*, 15 March 2013.

Distribution

10	MS0421	Edward R. Carroll	00158
1	MS0421	Daniel Briand	00158
1	MS0421	Brad M. Mickelsen	00158
1	MS0421	Systems Analysis Tech Lib	00150
1	MS0453	Brett L. Remund	02210
1	MS0623	Tamara S. Rodriquez	02215
1	MS0623	Whitney D. Wolf	02214
1	MS0623	Jeanne L. Evans	02214
1	MS0661	Jason P. Jarosz	02627
1	MS1026	Shawn M. Dirk	02241
1	MS1026	Sharon L. Trauth	02241
1	MS1138	Marjorie T. McCormack	06923
1	MS1138	Anthony R. Matta	06923
1	MS0899	Technical Library	9536 (electronic copy)

Sandia National Laboratories