2016-09

# Evaluation of Model-Based Systems Engineering processes for integration into rapid acquisition programs

García, Ramón A.

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/50550

# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**EVALUATION OF MODEL-BASED SYSTEMS ENGINEERING PROCESSES FOR INTEGRATION INTO RAPID ACQUISITION PROGRAMS**

by

Ramón A. García

September 2016

| | |
|---|---|
| Thesis Advisor: | Rama Gehris |
| Second Reader: | Gregory Miller |

THIS PAGE INTENTIONALLY LEFT BLANK

| | | | |
|---|---|---|---|
| **REPORT DOCUMENTATION PAGE** | | | *Form Approved OMB No. 0704–0188* |
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503. | | | |
| **1. AGENCY USE ONLY** ***(Leave blank)*** | **2. REPORT DATE**<br>September 2016 | **3. REPORT TYPE AND DATES COVERED**<br>Master's thesis | |
| **4. TITLE AND SUBTITLE**<br>EVALUATION OF MODEL-BASED SYSTEMS ENGINEERING PROCESSES FOR INTEGRATION INTO RAPID ACQUISITION PROGRAMS | | | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** Ramón A. García | | | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | | | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)**<br>N/A | | | **10. SPONSORING/ MONITORING AGENCY REPORT NUMBER** |
| **11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____. | | | |
| **12a. DISTRIBUTION/AVAILABILITY STATEMENT**<br>Approved for public release. Distribution is unlimited. | | | **12b. DISTRIBUTION CODE** |

**13. ABSTRACT (maximum 200 words)**

This thesis describes a framework upon which programs, particularly those identified as engaging in rapid acquisition, can compare themselves to determine if the adoption of a Model-Based Systems Engineering (MBSE) approach might be feasible. The research was established as a case study of several defense acquisition programs that are using MBSE as part of their software development process by providing a background for those programs being evaluated, then delving into their individual MBSE processes to identify the principal elements that added the most value in terms of delivering a suitable and effective product expediently. After completing the characterization of the MBSE approaches, an assessment of a sample target program was conducted, exercising the framework developed. The research indicates that while the implementation of MBSE can require additional effort during the initial development stages, the demonstrated benefits typically outweigh the extra upfront burden by reducing the overall design cycle time and improving the validation and verification activities. An in-depth mapping of the upfront MBSE work required would provide additional engineering rationale to justify the programmatic investment for implementing an MBSE approach.

| | | | |
|---|---|---|---|
| **14. SUBJECT TERMS**<br>MBSE, rapid acquisition, software development process, analysis framework, evaluation guidelines | | | **15. NUMBER OF PAGES**<br>77 |
| | | | **16. PRICE CODE** |
| **17. SECURITY CLASSIFICATION OF REPORT**<br>Unclassified | **18. SECURITY CLASSIFICATION OF THIS PAGE**<br>Unclassified | **19. SECURITY CLASSIFICATION OF ABSTRACT**<br>Unclassified | **20. LIMITATION OF ABSTRACT**<br>UU |

THIS PAGE INTENTIONALLY LEFT BLANK

# EVALUATION OF MODEL-BASED SYSTEMS ENGINEERING PROCESSES FOR INTEGRATION INTO RAPID ACQUISITION PROGRAMS

Ramón A. García
Civilian, Naval Undersea Warfare Center Division, Newport, Rhode Island
B.S., Yale University, 1990
M.S., Tufts University, 1993

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN SYSTEMS ENGINEERING MANAGEMENT**

from the

**NAVAL POSTGRADUATE SCHOOL**
**September 2016**

Approved by: Rama Gehris, Ph.D.
Thesis Advisor

Gregory Miller
Second Reader

Ronald Giachetti, Ph.D.
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

This thesis describes a framework upon which programs, particularly those identified as engaging in rapid acquisition, can compare themselves to determine if the adoption of a Model-Based Systems Engineering (MBSE) approach might be feasible. The research was established as a case study of several defense acquisition programs that are using MBSE as part of their software development process by providing a background for those programs being evaluated, then delving into their individual MBSE processes to identify the principal elements that added the most value in terms of delivering a suitable and effective product expediently. After completing the characterization of the MBSE approaches, an assessment of a sample target program was conducted, exercising the framework developed. The research indicates that while the implementation of MBSE can require additional effort during the initial development stages, the demonstrated benefits typically outweigh the extra upfront burden by reducing the overall design cycle time and improving the validation and verification activities. An in-depth mapping of the upfront MBSE work required would provide additional engineering rationale to justify the programmatic investment for implementing an MBSE approach.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

# LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| ACB | Advanced Capability Build |
| APB | Advanced Processing Build |
| ARCI | Acoustic Rapid COTS Insertion |
| ASIC | Application-Specific Integrated Circuits |
| ASW | Anti-Submarine Warfare |
| CASE | Computer-Aided Systems Engineering |
| CDD | Capabilities Development Document |
| CONOPS | concept of operations |
| COTS | commercial-off-the-shelf |
| CRUDES | cruisers and destroyers |
| CSCI | Computer Software Configuration Items |
| DAU | Defense Acquisition University |
| DOD | Department of Defense |
| EA | Electronics Assembly |
| EDM | Engineering Development Model |
| FBM | Fleet Ballistic Missile |
| HW | hardware |
| HWIL | hardware-in-the-loop |
| IDL | Interface Descriptive Language |
| IMU | Inertial Measurement Unit |
| INCOSE | International Council on Systems Engineering |
| MBSE | Model-Based Systems Engineering |
| MTBF | Mean Time Between Failure |
| MTBCF | Mean Time Between Critical Failure |
| MIRV | Multiple Independently-targetable Reentry Vehicle |
| MK6LE | MK6 Guidance System Life Extension |
| NUWC | Naval Undersea Warfare Center |
| PEO-IWS5 | Program Executive Office—Integrated Warfare Systems 5, Undersea Systems |
| PEO-SUBS | Program Executive Office—Submarines |

| | |
|---|---|
| ROC | Receiver Operating Characteristic |
| RTL | Register-Transfer Level |
| SE | systems engineering |
| SW | software |
| SysML | Systems Modeling Language |
| TI | Technical Insertion |
| TRL | Technology Readiness Level |
| UML | Unified Modeling Language |
| USN | United States Navy |
| USW | Undersea Warfare |
| VSSim | Virtual System Simulation |

# EXECUTIVE SUMMARY

Many Department of Defense (DOD) acquisition program offices have adopted aggressive development strategies to deliver systems to the warfighter in a more expedient fashion. These strategies have resulted in the execution of programs using rapid acquisition processes. However, as the acquisition cycles are shortened and budgets become more constrained, it is essential to identify new methods for system development. With the increasing rate of technological change and complexity in systems, the application of systems engineering (SE) principles is becoming more vital, making Model-Based Systems Engineering (MBSE) a more favorable alternative than "the traditional document-based SE approach" (Steward 2015, 1). The use of an MBSE approach can be the differentiator to reduce the design cycle time and provide better products at reduced costs.

This thesis establishes a framework upon which to evaluate different MBSE approaches to assess how they might be integrated into the development processes of rapid acquisition programs. The framework executes the following steps: (a) the establishment of the evaluation parameters, (b) the comparison of the selected MBSE processes, (c) the identification of the principal implementation factors, as well as the benefits and key enablers of the different MBSE approaches, and (d) the evaluation of the target program. The final step of this assessment process where the evaluation is conducted uses the main characteristics from the target program, viewed through the lens of the selected criteria, as a litmus test for performing parameter tradeoffs for the key MBSE enablers. By providing guidelines for evaluation, this research helps to establish a basis for the target projects to achieve greater success through the use of MBSE.

As different projects may have diverging concerns, it was necessary to select a specific program to provide a context filter to evaluate the different processes and to limit the comparison trade space. The target program selected was the SQQ89 due to the author's familiarity with the program, applicability as a target program due to its rapid acquisition nature, and opportunity for study. While the cost-effective quality of the

SQQ89 has been documented (Wilson 2009), the program has not currently adopted an MBSE approach.

This research evaluated three separate defense acquisition programs currently using MBSE methods as part of their software development efforts. These programs were the MK54 Lightweight Torpedo program, a Raytheon Radar program, and the Life Extension of the MK6 Guidance System (MK6LE) of the Trident II D5 Missile program. The respective MBSE approaches were captured and characterized in detail, then compared and contrasted to bring out their intrinsic properties.

The output result of the characterization of the MBSE approaches was a list of benefits gained by each project through the use of MBSE. In addition, a set of enablers that had a significant impact toward making the projects successful were identified. The capture of the key MBSE enablers represents an important part of this study, as it helps to communicate certain aspects of the MBSE development approaches that should be replicated industry wide. These enablers, along with selected evaluation parameters, set the foundation for the assessment of the target program.

Evaluation parameters were selected to assess the development processes, with the final criteria chosen for assessment being requirements maturity, scope clearness, stakeholder commitment, development stability, tool availability, tool supportability, process flow definition, and ease of implementation.

This research concluded with an evaluation of the target program by taking into account the main characteristics from the SQQ89 in the context of the selected criteria and the implementation factors identified. The assessment conducted addressed the research questions listed as follows, along with a brief summary of the answers obtained:

> **What are the pros and cons of each of the processes?**
>
> In general, the main drawback was the initial effort required to establish an MBSE approach. The use of MBSE requires not only engineering rigor but also programmatic commitment. However, once implemented, an MBSE design approach was invaluable for the three programs that were studied, as it supported each project’s development goals. MBSE efforts focused not only on the early system requirements, design, and analysis phases but also the verification and

validation activities throughout the later life-cycle phases. MBSE allowed the programs to manage the evolution of simulation capabilities, as well as to assess the appropriate fidelity required to meet development needs.

**What elements of each process add the most value to their project?**

The three programs shared some common enablers that added the most value: (a) organizational cachet, (b) stable high-level requirements, and (c) clearly defined interfaces. The organizational cachet gave the programs the confidence to embark with the using MBSE. While not restricted to MBSE programs alone, clarity of purpose, regarding the stability of requirements, interfaces, and approach were seen to contribute greatly to project success. Careful planning, supported by a holistic MBSE approach, brought about some project-unique enablers to each process. Whether leveraging their access to historical data, reuse of system design tools, or the embedding of the Modeling and Simulation team into the design effort, these intrinsic elements were used to remove developmental "stovepipes." By exploiting all the capabilities of an MBSE approach, from design to validation, the programs were able to meet their development milestones successfully within the planned timelines.

**What attributes of the rapid acquisition projects might be improved from implementing an MBSE process?**

MBSE in general was identified as providing better quality requirements, resulting in lower rework. Combined with the gains achieved to the significant labor reduction due to automation, the MBSE approach provided improvements to the programs quality, schedule and cost. By providing repeatable test vectors with the required fidelity, the confidence levels associated with the designs was increased. The ability to automate testing and increase the test coverage allowed for a better assessment of model performance for existing functionality, as well as improving development and validation of new algorithms. Overall, the use of an MBSE approach helped improve the redesign, supportability and suitability of the programs reviewed.

Implementation of MBSE can require additional effort during the initial development stages but has demonstrated benefits, such as the reduction in design defects and increased capability to verify requirements, which typically outweigh the extra upfront burden by reducing the overall design cycle time. As presented through the discussion of the different case studies, the establishment of an MBSE approach requires programmatic commitment from the customer. There is an initial level of inertia and fear that needs to be overcome, but once commitment exists toward this investment, the payoff can be great.

Through the use of MBSE, the traceability of requirements was improved since models developed at the system, subsystem, and CSCI levels could directly capture capabilities, functions, and requirements in a form that traced back directly to the customer needs. The visualization of the current approaches to MBSE is closing the gaps between the simulation and the actual systems themselves. As this coupling between the system models and application software becomes ever tighter, the line between the simulation and executable domains is being blurred.

Rapid acquisition development demands high confidence. As was evidenced with the programs evaluated, designs could be quickly iterated so that they worked the first time out of the box. In fact, in all cases evaluated, MBSE resulted in an order of magnitude in time compression.

Based on this framework, it is the author's opinion that incorporation of MBSE processes should be recommended for the SQQ89. The scale, timelines, development paradigm, and verification needs of the SQQ89 program serve to justify the effort to engage in MBSE approach. The use of an MBSE approach provides palpable advantages from a technical and engineering standpoint. Nonetheless, it may be the programmatic benefits in terms of reduced time and cost that are needed to get customer buy-in of an MBSE approach. These advantages would certainly benefit current rapid acquisition programs, such as the SQQ89.

This research starts to create a framework to identify the fit between the programs and the MBSE approach. An additional step that would be helpful toward this process would be to generate an in-depth mapping of the upfront MBSE work required. A detailed cost-benefit analysis would provide additional engineering rationale for embarking upon the use of MBSE. The follow-on assessment would serve to establish a business case to further justify the programmatic investment and overcome the fear of implementing an MBSE approach.

## REFERENCES

Steward, Victoria. 2015. “Functional Flow and Event-Driven Methods for Predicting System Performance.” Master’s thesis, Naval Postgraduate School.

Wilson, Anthony M. 2009. “The Impact of Software Reuse on the Cost of Navy Sonar and Fire Control Systems.” Master’s thesis, Naval Postgraduate School.

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

I sincerely thank all the individuals who provided me with their encouragement and support throughout the process of writing this thesis. Without them, this research could not have been completed. Special thanks go to:

- Jeff Cipolloni, Dan Keating, and Scott Dellicker from Draper. Their support of the NPS program provided the initial spark for this thesis. It is my hope that this work allows for a fruitful collaboration that will extend far beyond this research.
- Phil Campo, Bill Jones, Kevin Chaves, and Steve Bordonaro from the Naval Undersea Warfare Center. Thanks to all for their willingness to help and their interest in this work.
- Chris Finlay from Raytheon Company, for sharing his MBSE insight.
- My advisor, Rama Gehris, and second reader, Gregory Miller, for sharing their expertise, guidance, and assistance to help me finish on time.

Finally, I would like to thank my family for their love and support. They had to endure a lot through the last two years of graduate work, but the journey is finally over. And last but certainly not least, I would like to thank my mom, Marieselle Serrano, for her never-ending faith in me that I would be able to see it all through.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

## A. PROJECT OVERVIEW AND GOALS

This thesis provides an independent look across several defense acquisition programs using Model-Based Systems Engineering (MBSE) as part of their software (SW) development approach. Each program's MBSE process is mapped and evaluated with the goal of identifying the principal elements within the respective programs that add the most value in terms of delivering a suitable and effective product expediently. The aim of this work is to characterize those elements for inclusion into other SW development programs, serving as a benchmark for best practice activities, or at a minimum to identify lessons learned for programs that are considering implementing MBSE as part of their SW development process.

The ultimate goal of this research is to provide a framework upon which programs, particularly those identified as engaging in rapid acquisition, can be evaluated to determine if the adoption of an MBSE approach might be feasible. This framework consists of a methodology to perform qualitative assessments, conducted in the context of a desired end state and compared against the main characteristics of the target program being evaluated.

This research evaluated three separate programs currently using MBSE methods. A case-study approach was used to gain the background for the programs being evaluated, to delve into their individual processes, and to extract key characteristics of each MBSE approach. The output result is a list of benefits gained by each project through the use of MBSE and a set of enablers that were identified as making the projects successful. These enablers, along with selected evaluation parameters, set the foundation for the assessment of the target program. By providing guidelines for evaluation, this research helps to establish a basis for the target projects to achieve greater success through the use of MBSE.

## B. OVERVIEW OF MBSE AND RAPID ACQUISITION

Starting in the year 2001, the Department of Defense (DOD) changed the basis for how acquisition programs define requirements (Walker 2005). Instead of using a “threat-based” process to define requirements for system acquisitions, a “capabilities-based” model was adopted (Rumsfeld 2001, iv). The intent of the capabilities-based approach was to derive requirements in a top-down approach in order to meet the intended operational need.

Many program offices have adopted aggressive development strategies to deliver systems to the warfighter in a more expedient fashion. These strategies have resulted in execution of programs using rapid acquisition processes. Rapid acquisition has been defined as

> a contractually obligated acquisition effort that, relatively speaking, attempts to complete either a typical amount of “acquisition product” on a compressed schedule or an above average amount of “acquisition product” on a typical schedule. Typical here is a subjective judgment for what appears to be standard for the industry… For example, it is not typical to deliver flight hardware in two years. This would be considered an aggressive schedule. (Johnson 2010)

As the acquisition cycles are shortened and budgets become more constrained, it is essential to identify new methods for system development.

### 1. TRENDS FOR EXECUTING RAPID ACQUISITION

In the late 1990s, the Program Executive Office for Submarines (PEO-SUBS) initiated the Acoustic Rapid commercial-off-the-shelf (COTS) Insertion (ARCI) program. The ARCI program “migrated various submarine-specific sonar and fire control systems away from military-unique computers” and transitioned them “to combat systems written in modern programming languages running on modern commercial processors” as indicated by Mitchell (2014, 316).

According to Wilson (2009), ARCI sought “to improve sonar systems in the submarine force through the use of commercial technology and planned upgrades to take advantage of advances in technology” (9). Wilson also indicates that this was done with the goal “to leverage advances in commercial technology to provide leading-edge products to fleet end users.” (9) More specifically, Mitchell (2014) notes that ARCI “migrated various class-unique submarine sonar and fire control systems away from military-unique computers such as the AN/UYQ-7 and AN/YUK-43” (316). These military-grade products typically took up to 10 years to design, fabricate, and field. According to Wilson (2009), “the first ARCI upgrade was provided to the fleet in November of 1997,” 18 months after program startup (9).

The near doubling in integrated circuit complexity as defined by Moore’s law (Brock and Moore 2006) has led to annual increases in computing power and processing speed for less cost. Accordingly, through the employment of a rapid acquisition strategy that reduced the time to field products, the Navy was able to exploit these exponential gains exhibited by COTS products to provide better performance and capabilities as they became available (Wilson 2009).

Building upon the approach used by the ARCI program, the Advanced Processing Build (APB) development for the Los Angeles 688-class fast attack submarine platforms expanded the scope from a hardware (HW) focus to include SW elements. This combined evolutionary approach to HW and SW development has been recently replicated by rapid acquisition programs such as the AN/SQQ-89. Initially deployed in the Spruance DD963-class (Global Security, 2011) to support the Anti-Submarine Warfare (ASW) mission, the AN/SQQ-89 consists of acoustic detection sensors, information processing, and combat control capabilities designed to detect, classify, localize, and engage undersea threats.

Currently, the AN/SQQ-89 is the Undersea Warfare (USW) Combat System for AEGIS Ticonderoga CG47-class cruisers and Arleigh Burke DDG51-class destroyers (CRUDES). Led by the Program Executive Office—Integrated Warfare Systems 5, Undersea Systems (PEO-IWS5) organization, the AN/SQQ-89 employs a variant of the incremental development paradigm, geared toward maximizing SW re-use and HW commonality across the CRUDES platforms. For the context of this research, the term “‘SQQ89” will be used in reference to the latest variants of the AN/SQQ-89 known as A(V)15 systems (Figure 1).

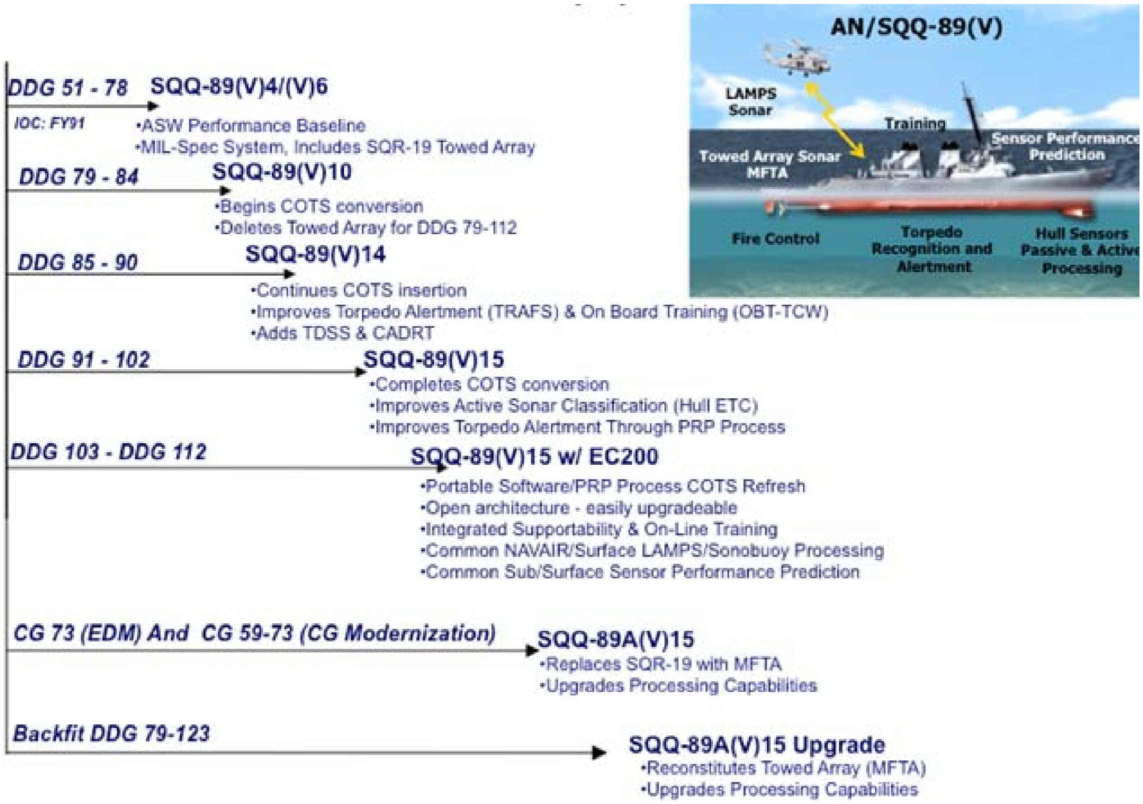
AN/SQQ-89(V)
LAMPS Sonar
Training
Sensor Performance Prediction
Towed Array Sonar MFTA
Fire Control
Torpedo Recognition and Alertment
Hull Sensors Passive & Active Processing
DDG 51 - 78
SQQ-89(V)4/(V)6
IOC: FY91
•ASW Performance Baseline
•MIL-Spec System, Includes SQR-19 Towed Array
DDG 79 - 84
SQQ-89(V)10
•Begins COTS conversion
•Deletes Towed Array for DDG 79-112
DDG 85 - 90
SQQ-89(V)14
•Continues COTS insertion
•Improves Torpedo Alertment (TRAFS) & On Board Training (OBT-TCW)
•Adds TDSS & CADRT
DDG 91 - 102
SQQ-89(V)15
•Completes COTS conversion
•Improves Active Sonar Classification (Hull ETC)
•Improves Torpedo Alertment Through PRP Process
DDG 103 - DDG 112
SQQ-89(V)15 w/ EC200
•Portable Software/PRP Process COTS Refresh
•Open architecture - easily upgradeable
•Integrated Supportability & On-Line Training
•Common NAVAIR/Surface LAMPS/Sonobuoy Processing
•Common Sub/Surface Sensor Performance Prediction
CG 73 (EDM) And CG 59-73 (CG Modernization)
SQQ-89A(V)15
•Replaces SQR-19 with MFTA
•Upgrades Processing Capabilities
Backfit DDG 79-123
SQQ-89A(V)15 Upgrade
•Reconstitutes Towed Array (MFTA)
•Upgrades Processing Capabilities

Figure 1. AN/SQQ-89(V) Evolution. Adapted from Armstrong (2007).

Using a procurement and development strategy that combines elements from the APB SW (PEO-IWS5 2003) and ARCI HW production (Johnson 2004) paradigms, the SQQ89 incorporates new performance improvements through a process currently described as the Advanced Capability Build (ACB) process. The ACB process integrates new SW improvements, making them available for installation on a two-year cycle, along with a Technical Insertion (TI) of HW, staggered by one year. This one-year gap allows the software to be developed for a new HW set, then extends the system's supportability by enabling the same HW set to support the next ACB SW upgrade. After two years, the introduction of the next HW set initiates the next cycle (Figure 2). For example, the most recently fielded ACB SW baseline was delivered in 2011 (ACB11) with TI-12 HW. The follow-on certified build of ACB SW (ACB13) also runs on TI-12 HW, providing a SW upgrade path for that HW configuration. In this way, technology improvements are targeted for upcoming ACBs, creating a development roadmap that allows for the planning of future builds based on existing capability gaps or emerging needs.

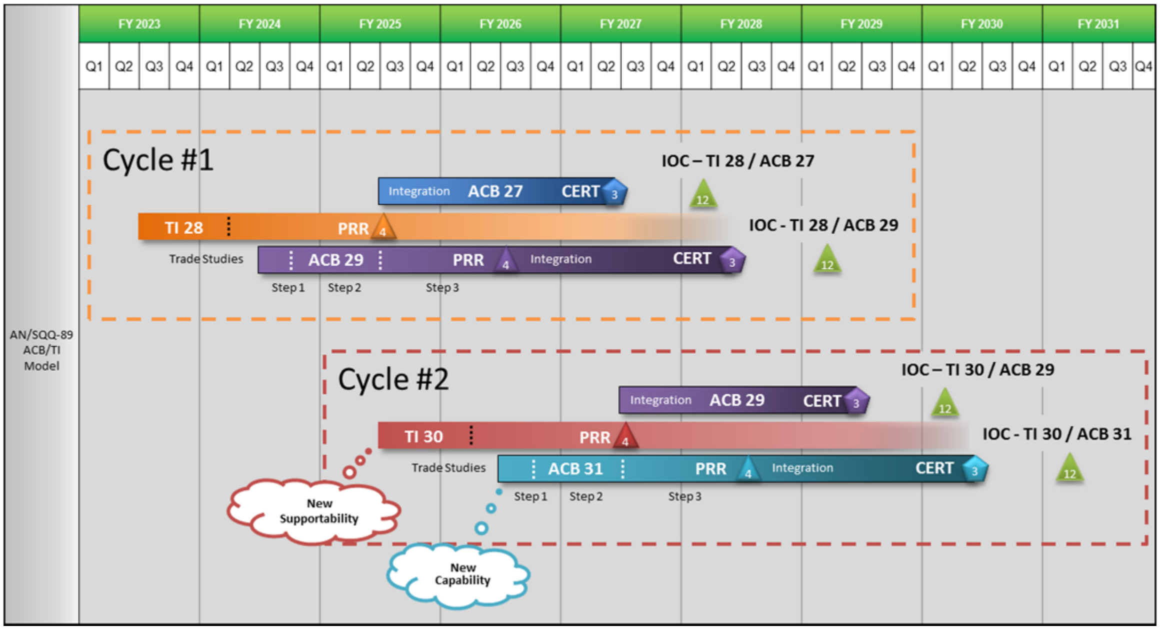

Figure 2. Conceptual TI/ACB Development and Integration Business Model. Source: PEO-IWS5 (2016).

Similarly to the submarine APB process, the ACB process (Figure 3) introduces new capabilities to future SW baselines incrementally. The capability development takes place in phases or steps by evaluating the maturity of the product at each step, and only transitioning forward the capabilities that meet predetermined exit criteria. The five development steps are described as follows:

Step 1: Technology Evaluation. Working groups consider products developed by industry, the Navy, and other DOD agencies to determine their tactical utility, maturity, expected performance, and computational resource requirements.

Step 2: Algorithm Assessment. Mature technologies, vetted through Step 1, are implemented in a laboratory-computing environment through unit code and subjected to commonly available (open) recorded at-sea data.

Step 3: System Real Time Evaluation. Technologies that transition out of Step 2 are integrated into the tactical systems to ensure they can provide increased performance. Additional evaluations with reserved or closed data sets are also conducted to assess system effectiveness.

Step 4: At-Sea Testing. The integrated system is temporarily installed on the intended platform to evaluate its performance in an operationally realistic environment.

Step 5: Production. Capabilities that transition from Step 4 are formally incorporated into the tactical production baseline, where they complete the required verification and certification activities before being introduced into the surface fleet.

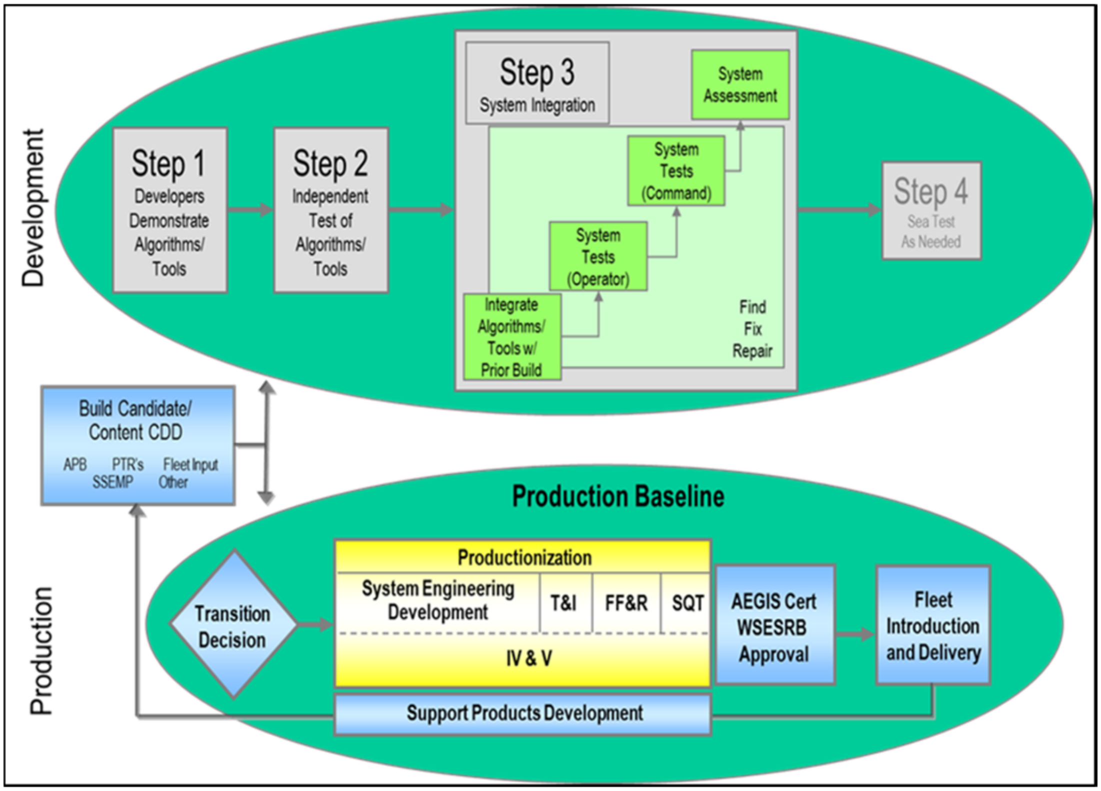
Development
Step 1
Developers Demonstrate Algorithms/ Tools
Step 2
Independent Test of Algorithms/ Tools
Step 3
System Integration
System Assessment
System Tests (Command)
System Tests (Operator)
Integrate Algorithms/ Tools w/ Prior Build
Find Fix Repair
Step 4
Sea Test As Needed
Build Candidate/ Content CDD
APB
PTR's
Fleet Input
SSEMP
Other
Production Baseline
Production
Transition Decision
Productionization
System Engineering Development
T&I
FF&R
SQT
IV & V
AEGIS Cert WSESRB Approval
Fleet Introduction and Delivery
Support Products Development

Figure 3. ASW Advanced Capability Build Development and Production Processes. Source: PEO-IWS5 (2016).

The step process has become a standard part of the PEO-IWS5 development strategy, and has been documented in multiple sources (PEO-IWS5 2003, PEO-IWS5 2016). However, the use of MBSE is currently very limited in scope within the ACB process. The following describes several features of the step process that would warrant incorporating an MBSE approach:

- a short (two-year) delivery cycle

- the ability to support multiple HW sets
- the integration of new capabilities seamlessly into an existing software baseline or architecture
- the ability to effectively assess performance at different stages of development

This research strives to determine if the adoption of an MBSE approach might be feasible within the ACB process.

### 2. THE DRIVE TOWARD MODEL-BASED SYSTEMS ENGINEERING

Engineering models have been described as “a system which we use as a surrogate” that “simplify or abstract the real system to reduce cost and/or focus on essential characteristics” (Sanchez 2007, 1). As described by Baker et al. (2000), “system engineers build models to better understand problems, develop candidate solutions, and validate their decisions” (1). Buede (2000) describes models as “abstractions of reality” which “start as very high level representations” and “become more detailed mathematical and sometimes physical representations of the system as the design portion of the development phase ends” (59).

However, current approaches to MBSE are closing the gap between the simulation and the actual systems themselves. Model-Based Systems Engineering has been defined as

> the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. (INCOSE 2007)

With the increasing rate of technological change and complexity in systems, the application of systems engineering (SE) principles is becoming more vital. The increase in complexity, coupled with the drive to deliver systems at a faster pace, has made the implementation of MBSE a more favorable alternative than “the traditional document-based SE process” (Steward 2015, 1).

According to Tepper (2010), the use of an MBSE approach can “provide a viable way to design with capabilities-based requirements in a solution neutral context” (15). In the same study, Tepper (2010) also points out how “MBSE provides the system designer a rigorous means for capturing and integrating system requirements, design, analysis, and verification information” (35). Similarly, the organization MBSE.Works identifies reasons for using an MBSE approach, namely to

- facilitate communication among various stakeholders across the system development life cycle
- capture and manage corporate intellectual property related to system architectures, designs, and processes
- compare and contrast “as is” and “to be” solutions
- explore multiple solutions or ideas concurrently with minimal risk
- detect errors and omissions early in the System Development Life cycle (MBSE.Works 2015)

Modeling, and MBSE in general, has been identified (Ramos et al. 2012) to provide potential benefits such as “improved communications,” and “consistent and complete representations of a system” (108). With decreasing budgets and increasing global threats, there is a need for DOD rapid acquisition programs to explore efficiencies not currently achievable through traditional development methods. The use of an MBSE approach can be the differentiator to reduce the design cycle time, and provide better products at reduced costs.

## C. LITERATURE REVIEW

The goal of the literature review was to gain knowledge and understanding of MBSE, capabilities-based systems development, rapid acquisition efforts, modeling and simulation, application of MBSE to the architecting and development of systems, as well as existing methods and practices that have been used to perform comparative as well as interpretive evaluations.

Research into recent capabilities-based efforts revolves around the use of MBSE, as it provides a rigorous way to capture and build upon system requirements to develop and evaluate conceptual architectures. Addington (2008) uses a model-based methodology for capabilities-based systems development, or more specifically, for system of systems (SoS) architecture development. Letourneau (2009) introduces a capabilities-driven SE process as part of an effort to incorporate multi-criteria optimization and uncertainty analysis, when applied to an autonomous surface craft architecture. Tepper (2010) covers a capabilities-based approach, using MBSE to develop a system architecture in the context of Navy ship design and acquisition, and Giammarco (2012) proposes the application of formal methods to architecture model quality assessment.

Additional discussion regarding the use of MBSE further into the design phase through product development stages is introduced by Fitzgerald et al. (2000) and Sandhu (2015). Fitzgerald et al. focus on the tailoring of system development methods, using a multi-level (industry, organizational, and project level) tailoring process. Sandhu (2015) covers different types of MBSE approaches, based on their primary abstraction levels, described as (a) “model centric software development,” (b) “model driven development,” (c) “architecture centric development,” (d) “specification-driven development,” and (e) “generative and component-based approaches” (1842).

Topper (2013) discusses the use of MBSE concepts and techniques in the development of complex systems. Bahlman (2012) applies an MBSE approach to ship design, to assess the “logical behavior and mission effectiveness” (v) through model evaluation. These works demonstrate how MBSE can add value to an overall design process.

Insights into measures and techniques for evaluating MBSE processes were obtained from existing literature. Alexander and Davis (1991) provide an early look at guidelines for selecting “the most appropriate process models for particular a project” (521), by applying relative criteria ratings for each model. This work is notable for its contribution prior to the stage where software models were formally identified as being key components of MBSE efforts.

Estefan (2007) performs a survey of some of the leading MBSE methodologies currently used in industry. Similarly, Ramos, Ferreira, and Barceló (2012) discuss the current state of MBSE “standards, formalisms, available modeling languages, methodologies, and the major applications” (108). Their review of key methodology characteristics is useful in ascertaining differences between MBSE approaches.

While presenting a vision for the field of SE to the year 2020, the International Council on Systems Engineering (International Council on Systems Engineering [INCOSE]) identifies a list of attributes that can be used to assess systems (INCOSE 2007). The study lists the “set of attributes that were selected to compare systems observe trends in the way systems have changed over the past 25 years” (11). They are

1. Purpose, Scope & Capability—autonomy
2. Complexity—including components & interfaces
3. Systems of Systems
4. Technology—used in the system itself
5. Embedded Software and information processing
6. Role of Humans—as part of the system
7. Legacy System Composition (INCOSE 2007, 11)

Robinson et al. (2010) describe the case-specific MBSE methodology used to derive requirements for a new capability, resulting in the generation of a Capability Definition Documentation. As part of this effort, they present requirements for the system models being created, stated as (a) “determinism with formality,” (b) “understandability,” (c) “inclusion of system missions,” (d) “modeling of structure and behavior,” and (e) “possibility of verification support” (Robinson et al. 2010, 3–4). These five requirements are identified as necessary conditions embodied by the model so that an MBSE approach can be applied.

Demirci (2010) developed an approach to assess MBSE maturity in an organization’s process. The result is a methodology based on the creation of an “MBSE/ UML Maturity Model” that allows the reader to “to evaluate the quality and proficiency in usage of UML” (Demirci 2010, 4). While Demirci focused on the level of proficiency specific to the Unified Modeling Language (UML), his approach could be applied generically to other efforts to provide a qualitative assessment such as is done with the determination of technology readiness levels (TRL) in the Defense Acquisition Guidebook published by the Defense Acquisition University (DAU) (2013).

Sharon et al. (2010) introduce “a decision framework for selecting a suitable software Development Process” (34), identifying potential metrics, criteria, and approach for process evaluation. The final factors and their associated weighting were selected by averaging the responses from a survey conducted by the authors.

The methodologies presented in the published works discussed earlier helped shape this research. Through the application of a holistic or systems view, the different works demonstrated how model-based efforts could be used to solve difficult problems and to produce tangible benefits. Furthermore, the analytic approaches used for evaluation of disparate projects helped in the characterization of the MBSE processes being evaluated herein, and to shape the criteria used, forming the foundation for this thesis.

## D. RESEARCH SCOPE

The goal for this thesis was to establish a framework to evaluate different MBSE approaches to assess how they might be integrated into the development processes of rapid acquisition programs. The steps executed to achieve this goal were (a) the establishment of the evaluation parameters, (b) the comparison of the selected MBSE processes, (c) the identification of the principal implementation factors, as well as the benefits and key enablers of the different MBSE approaches, and (d) the evaluation of the target program. The final step of this assessment process where the evaluation is conducted uses the main characteristics from the target program, viewed through the lens of the selected criteria, as a litmus test for performing parameter tradeoffs for the key MBSE enablers.

### 1. Research Questions

This thesis evaluated three separate DOD SW development programs as case studies to answer the following research questions:

- What elements of each process add the most value to their project?
- What are the pros and cons of each of the processes?
- What attributes of the rapid acquisition projects might be improved from implementing an MBSE process?

### 2. Research Approach

As different projects may have diverging concerns, it was necessary to select a specific program to provide a context filter to evaluate the different processes and to limit the comparison trade space. The program selected was the SQQ89 due to the author’s familiarity with the program, applicability as a target program due to its rapid acquisition nature, and opportunity for study. While the cost-effective quality of the SQQ89 has been documented (Wilson 2009), the program has not currently adopted an MBSE approach, using only limited modeling as will be discussed herein.

The MBSE approaches were captured and characterized in detail, then compared and contrasted to bring out intrinsic properties and benefits of each. Evaluation criteria was selected to assess the development processes, concluding with an evaluation of the SQQ89 target program. This evaluation took into consideration key characteristics of the SQQ89’s development paradigm, in the context of the selected criteria and the implementation factors identified.

The capture of the key MBSE enablers represents an important part of this study, as it helps to communicate certain aspects of the MBSE development approaches that should be replicated industry wide. However, the main benefit of this research is the presentation of the guidelines by which other rapid acquisition programs may perform their own assessments. Although this research focuses on the SQQ89, serving as the target program of interest, it is this author’s belief that the recommendations made in regards to the SQQ89 program would be extensible to other projects. Many of the characteristics of the SQQ89’s development paradigm reflect those of other rapid acquisition programs. Consequently, the analysis and conclusions could be applied accordingly beyond the SQQ89 as well.

# II. SELECTED CASES

The three DOD SW development programs selected for comparison are listed in Table 1. While the executing organizations vary in size and type, the programs themselves encompass similar levels of size and complexity. The respective processes employed by these programs follow.

Table 1. Selected DOD Programs for Evaluation of MBSE Processes.

| Program | Lead Organization | Organization Type |
|---|---|---|
| MK54 Torpedo | NUWC | U. S. Navy full-spectrum research, development, test and evaluation, engineering and fleet support center |
| Radar | Raytheon | Major American defense contractor and industrial corporation |
| Trident | Draper | Independent not-for-profit research and development company |

## A. CASE 1—MK54 TORPEDO PROGRAM

The FY2012 report from the Director, Operational Test and Evaluation identifies the MK54 torpedo (Figure 4) as "the primary Anti-Submarine Warfare weapon used by U.S. Navy surface ships, fixed-wing aircraft, and helicopters" (DOT&E 2013). The MK 54 has also been advertised as

> designed to defeat diesel as well as nuclear-powered submarines in shallow water and open-ocean environments. The MK54 can be deployed

> from a surface ship, helicopter or fixed-wing aircraft to track, classify and attack underwater targets. It uses sophisticated processing algorithms to analyze the information, edit out false targets or countermeasures, and then pursue identified threats. (Raytheon 2015)

Figure 4. MK54 Torpedo. Source: USN (2013).

### 1. System Overview

According to the United States Navy (USN) Fact File (2013),

> The MK 54 Mod 0 Lightweight Torpedo integrates existing torpedo hardware and software from the MK 46, MK 50, and MK 48 torpedo programs with state-of-the-art commercial-off-the-shelf (COTS) digital signal-processing technology. It incorporates an advanced guidance and control (G&C) section employing COTS processing technologies and tactical software improvements to significantly increase shallow water counter-countermeasure capability at reduced life cycle costs. (United States Navy [USN] 2013)

The original weapon software design was built on an architecture structure from the early 1980s and was based on hardware constraints that no longer exist. Carrying forward over 30 years of updates and patches, the software had become difficult to understand and change, was very time consuming to test, and was difficult to model accurately. In order to address all these concerns, the MK54 program transitioned to an MBSE re-design.

The new MK54 sonar processing uses an expandable, open architecture approach to improve modularity, provide scalability, and better support test and evaluation. Additionally, this new approach improved weapon performance and enabled the support of multiple product lines.

### 2. Software Development Process

The MK54 program utilized a modified version of the “Vee” process (Figure 5) initially introduced by Forsberg and Mooz in 1991. The process employed a top-down identification of system functionality, combined with a Scrum development approach Schwaber 2004), that was then allocated to lower level components. Follow-on upgrades efforts to provide performance improvements have been executed using the same modified Vee process but within the scope and scheduling paradigm of an ACB upgrade process as was discussed previously for the SQQ89. All follow-on work has used the baseline MBSE design as its departure point.

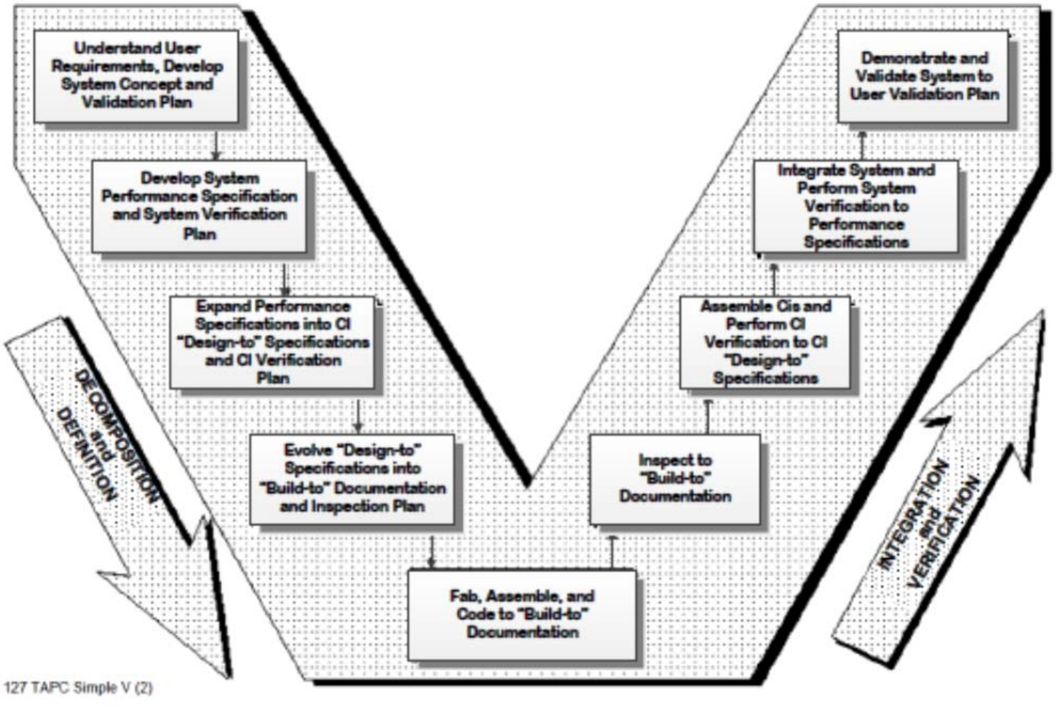
Understand User Requirements, Develop System Concept and Validation Plan
Develop System Performance Specification and System Verification Plan
Expand Performance Specifications into CI "Design-to" Specifications and CI Verification Plan
Evolve "Design-to" Specifications into "Build-to" Documentation and Inspection Plan
Fab, Assemble, and Code to "Build-to" Documentation
Inspect to "Build-to" Documentation
Assemble Cis and Perform CI Verification to CI "Design-to" Specifications
Integrate System and Perform System Verification to Performance Specifications
Demonstrate and Validate System to User Validation Plan
DE COMPOSITION and DEFINITION
INTEGRATION and VERIFICATION
127 TAPC Simple V (2)

Figure 5. Classic Systems Engineering "Vee" Model.
Source: Forsberg and Mooz (1991).

The SW development for the MK54 was guided by the legacy top-level Capabilities Development Document (CDD) down to the subsystems specifications, using the newly defined architecture (Figure 6) as the blueprint where principal design decisions were made. The development focused on the primary SW components, or Computer Software Configuration Items (CSCIs) by making them smaller than the predecessor SW modules, as well as ensuring the interfaces between them were clearly defined.

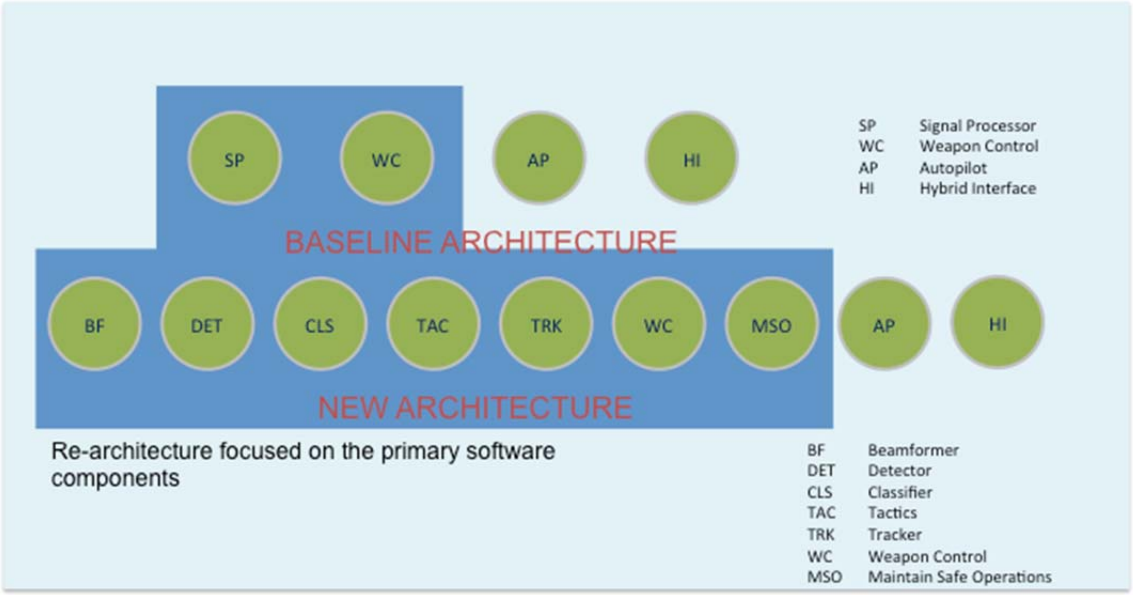

Figure 6. MK54 Re-Architecture/CSCI Design Approach.
Source: Bordonaro (2012).

### 3. MBSE Approach

The MK54 program utilized an MBSE-driven development to create a common torpedo development model. The MBSE design approach for this project (Figure 7) shows the different layers involved in the development. Written in the MATLAB language, the items in the application software layer comprise the core models for the torpedo. The resulting models exist at a level in which the operational code is isolated from the system hardware but fully capture the behavior at the subsystem level.

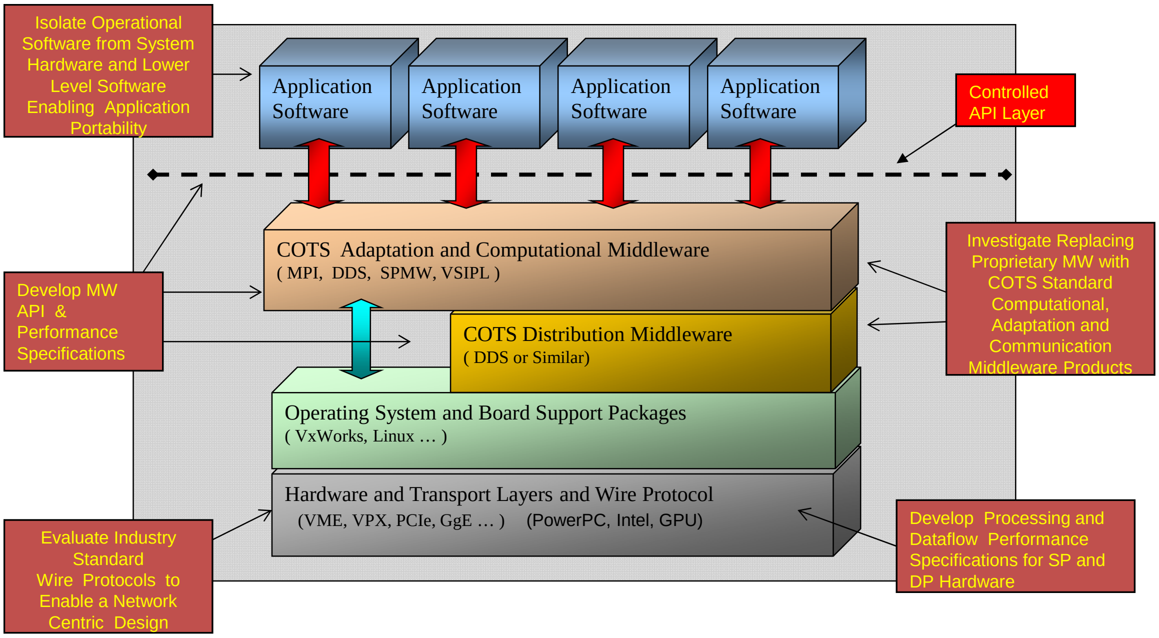
Isolate Operational Software from System Hardware and Lower Level Software Enabling Application Portability
Application Software
Application Software
Application Software
Application Software
Controlled API Layer
COTS Adaptation and Computational Middleware
( MPI, DDS, SPMW, VSIPL )
Investigate Replacing Proprietary MW with COTS Standard Computational, Adaptation and Communication Middleware Products
Develop MW API & Performance Specifications
COTS Distribution Middleware
( DDS or Similar)
Operating System and Board Support Packages
( VxWorks, Linux … )
Hardware and Transport Layers and Wire Protocol
(VME, VPX, PCIe, GgE … ) (PowerPC, Intel, GPU)
Evaluate Industry Standard Wire Protocols to Enable a Network Centric Design
Develop Processing and Dataflow Performance Specifications for SP and DP Hardware

Figure 7. MK54 MBSE Design Approach. Source: Bordonaro (2012).

The use of middleware products supports the abstraction of the SW from the HW layer. Using an open architecture construct, the SW development was not tied to any specific COTS implementation, and enabled the support of multiple product lines (torpedo variants) to achieve cost savings in algorithm development and maintenance. This approach led to the implementation of solution-neutral CSCIs that were dependent on the target missions, and not a specific vehicle on which to execute it.

A critical part in generating the CSCIs was a manual conversion from MATLAB to the C software language, so that the code could run in a real-time processing environment. As there were no automation tools for this conversion at the time, this effort became a critical step in the torpedo MBSE effort. The output product of this activity was the behavioral code that instantiated the torpedo functionality.

A key enabler in the MK54 development was the linkage in the data messages, through strategic management of the interfaces and data tap points. By allowing the models to interface with recorded at-sea data, a large set (over 10,000 hours) of historical inputs could be exercised. The use of actual data aided validation and verification efforts for the development and evolution of the models.

The ability to re-run test data facilitated metrics collection within the CSCIs, such as generation of Receiver Operating Characteristic (ROC) curves for the signal processor model, and classifier metrics “confusion matrix” for the classifier model. These characterizations could be done against specific builds or segment content. Interfacing the model with Analysis tools provided better visualization of torpedo performance, further improving system evaluation.

## B. CASE 2—RADAR PROGRAM

### 1. System Overview

In order to be compliant with publication requirements, the specific program details of the Raytheon Radar program were not divulged. Outside the general parameters of the project, involving a new developmental Radar intended to replace a legacy system, additional programmatic context was unavailable for unlimited distribution. However, the MBSE methodology was identified and is discussed next.

### 2. Software Development Process

Like the MK54 program, this Radar project utilized an agile Vee process, in a Scrum environment. However, in this case, the software development was guided by initial trade studies conducted as part of a proposal phase, and the build-up of an Engineering Development Model (EDM) as the initial representation of the customer’s requirements. As such, this effort morphed from an architecture-centric into a specification-driven development.

### 3. MBSE Approach

Once the initial customer requirements for the Radar were effectively identified, the MBSE process was invoked, using a tailored version of the IBM Harmony Process (Figure 8) described by Hoffman (2011). This tailored process started through the creation of use cases and epochs, based on warfighter missions and capabilities.

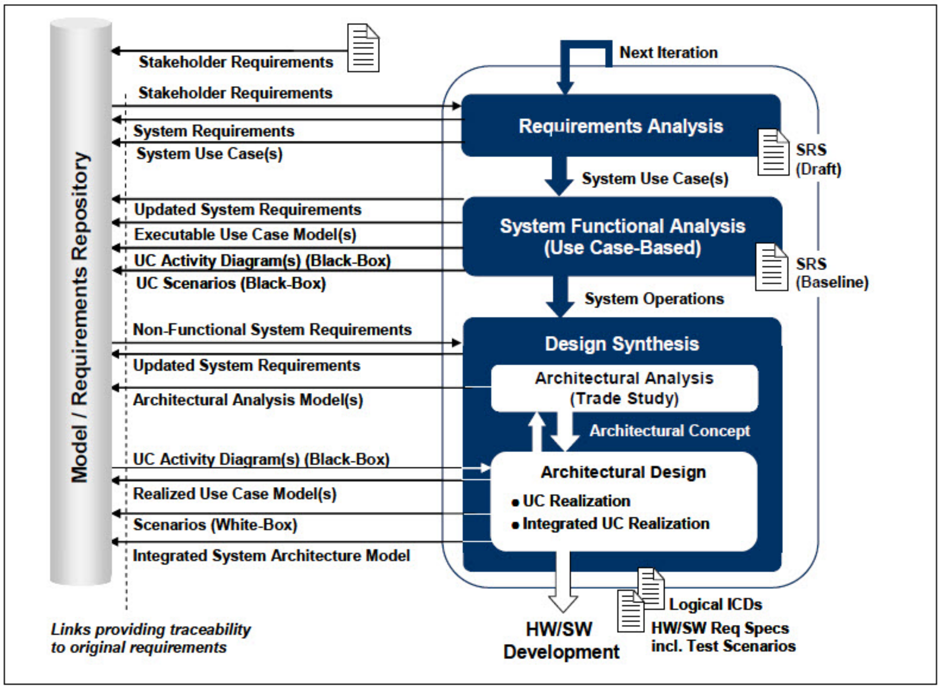
Model / Requirements Repository
Stakeholder Requirements
Stakeholder Requirements
System Requirements
System Use Case(s)
Updated System Requirements
Executable Use Case Model(s)
UC Activity Diagram(s) (Black-Box)
UC Scenarios (Black-Box)
Non-Functional System Requirements
Updated System Requirements
Architectural Analysis Model(s)
UC Activity Diagram(s) (Black-Box)
Realized Use Case Model(s)
Scenarios (White-Box)
Integrated System Architecture Model
Links providing traceability to original requirements
Next Iteration
Requirements Analysis
SRS (Draft)
System Use Case(s)
System Functional Analysis (Use Case-Based)
SRS (Baseline)
System Operations
Design Synthesis
Architectural Analysis (Trade Study)
Architectural Concept
Architectural Design
• UC Realization
• Integrated UC Realization
HW/SW Development
Logical ICDs
HW/SW Req Specs incl. Test Scenarios

Figure 8. IBM Harmony/Rational MBSE Development Model.
Source: Hoffman (2011).

The use cases were supplemented with activity diagrams to identify the functional flow, and sequence diagrams to define the system messages. From the activity diagrams, the lower level algorithms (sub-system activity and sequence diagrams) were then generated, iterating back up to the higher level as gaps or ambiguities are uncovered. Once this “base” model was created, an automated model validation check was executed by running the model against previously defined rule sets.

The Systems Modeling Language (SysML) version 1.2 was used to capture the architectural decomposition and interaction among elements and components, specifically the interface data model. By capturing the architectural decomposition, SysML served as the foundation for the broader MBSE environment.

Requirement linkage to Rational DOORS was done via Rhapsody Gateway, and automation was achieved through a Rhapsody plug-in development (leveraging the Rhapsody API). Once validated, the Rational Publishing Engine was used to auto-generate the Technical Data Package (such as Capability Functional Description

Documents, ICDs, and IDLs). This enabled the team to provide a turn-key system preliminary design, iterate as necessary to capture changes, and re-generate all the documentation within minutes.

For the Radar program, the models were implemented at the component/product level. The models generated, which are functional (behavioral) models, are all integrated through the Rhapsody Rational Publishing Engine. At this time, neither discrete performance (e.g., computer-aided design), HW, nor test models are linked to the overall system model. Only the requirements (DOORS), System, Interface and Data models are currently linked. The current state of model implementation as well as the desired (future) vision of an integrated system model is depicted in Figure 9.

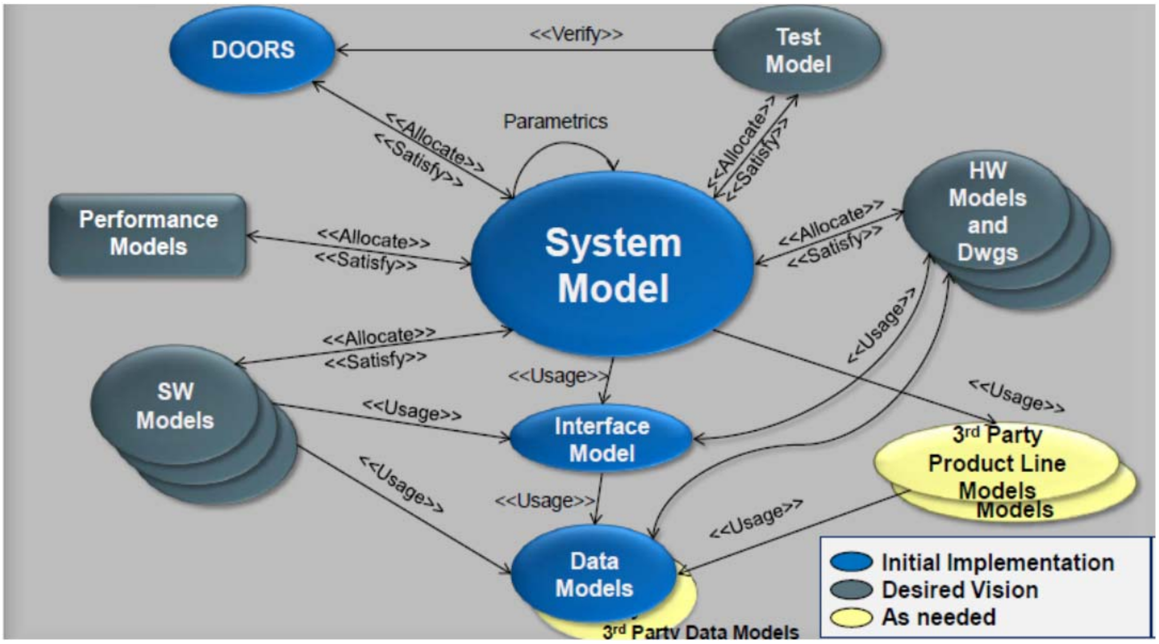

Figure 9. Depiction of the Integrated System Model. Source: Finlay and Dujardin (2014).

While the MBSE approach was able to auto-generate system design documentation, from models to Interface Descriptive Language (IDL), software team expertise was required to "dis-ambiguate" requirements and to convert the models to executable code. However, transitioning from an existing mix of document-based engineering to a common system model ensured consistency from requirements through detailed design, integration, test and life cycle support.

## C. CASE 3—TRIDENT MISSILE PROGRAM

The Trident II D5 is an "inertial guided ballistic missile" (Naval-Technology 2016). It is derived from the original Poseidon Fleet Ballistic Missile (FBM) System concept developed during the Cold War (Furhman 1978). A general description is provided by Lockheed Martin, SSC as follows

> First deployed in 1990, the D5 missile currently is aboard OHIO-class and British VANGUARD-class submarines. The three-stage, solid-propellant, inertial-guided ballistic missile can travel a nominal range of 4,000 nautical miles and carries multiple independently targeted reentry vehicles [MIRV] (Lockheed-Martin, SSC 2012)

Depictions of the MIRV (Figure 10) and the concept of operations (CONOPS) for the Trident II D5 missile (Figure 11) are provided.

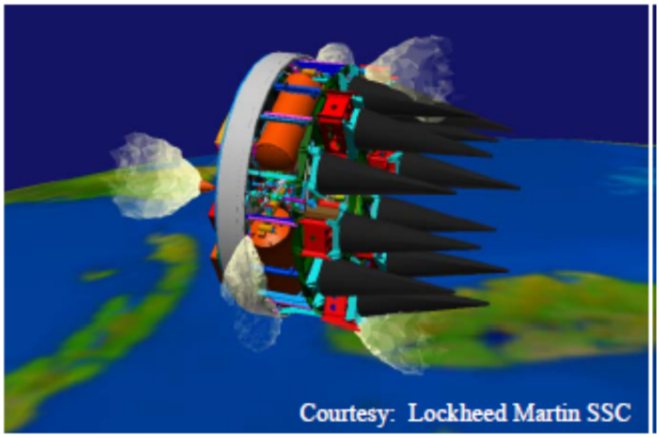

Figure 10. Multiple Independently Targetable Reentry Vehicle (MIRV). Source: Lockheed-Martin, SSC (2012).

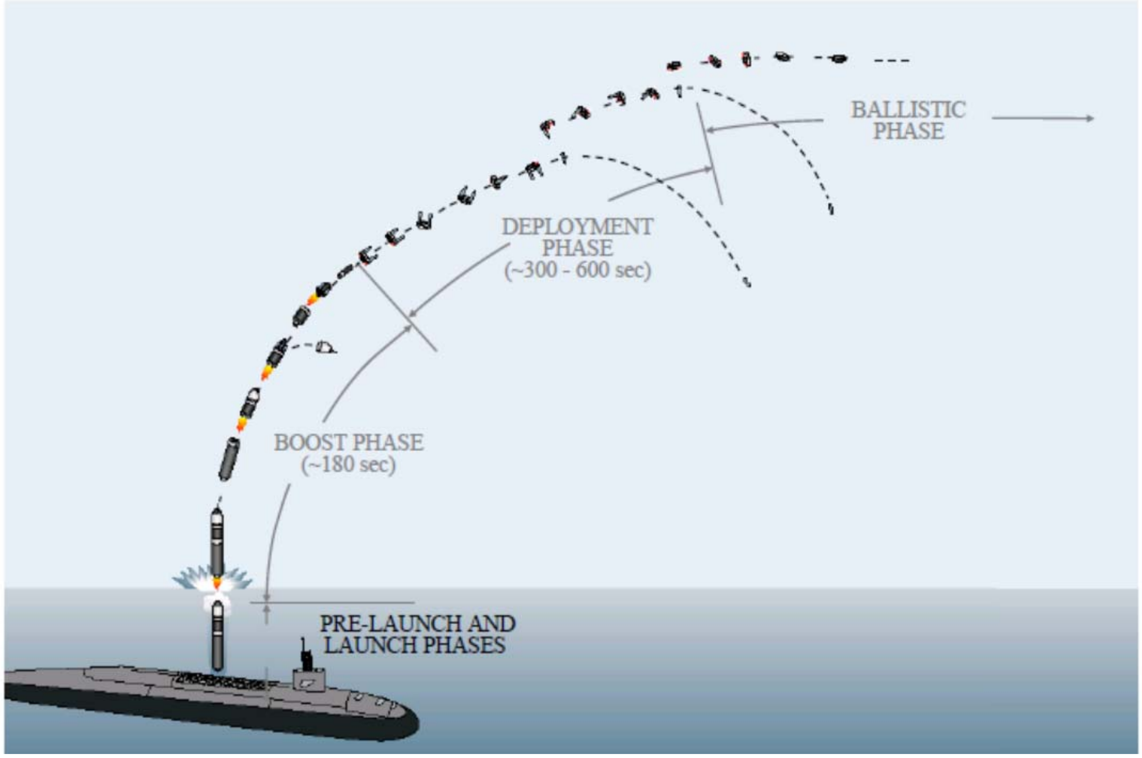
BALLISTIC PHASE
DEPLOYMENT PHASE (~300 - 600 sec)
BOOST PHASE (~180 sec)
PRE-LAUNCH AND LAUNCH PHASES

Figure 11. Trident II D5 Missile System CONOPS.
Source: Jackson, et al. (2008).

As reported by Naval-Technology

> The D5 Life Extension Programme was initiated by the Navy in 2002 to ensure high-level readiness and safety of nuclear deterrence capability of the submarine by replacing ageing components of the Trident II missile. (2012).

The specific part of the development effort utilizing an MBSE process that could be analyzed, again to be compliant with publication requirements, was the Life Extension of the MK6 Guidance System or MK6LE. The initial objective was to extend the service life of the MK6 Guidance System on through the year 2020 but was eventually extended to 2042, to coincide with the service life of the Ohio-class submarines.

The MK6LE is intended to "maintain demonstrated accuracy and reliability of the predecessor Trident system, meet all external coordinated interfaces and environments while allowing for mission adaptability and technology insertion" (Jackson et al. 2008, 4). The MK6LE is functionally decomposed into several major sections: the electro-mechanical subsystem, the velocity subsystem, the altitude measurement subsystem, the platform control subsystem, the mission subsystem, the communication and timing subsystem, the stellar subsystem, the input/output subsystem, and the power subsystem. These subsystems, and the components they operate (Figure 12), namely the inertial measurement unit (IMU) and electronics assembly (EA), as well as the physical decomposition of the MK6LE (Figure 13) are depicted.

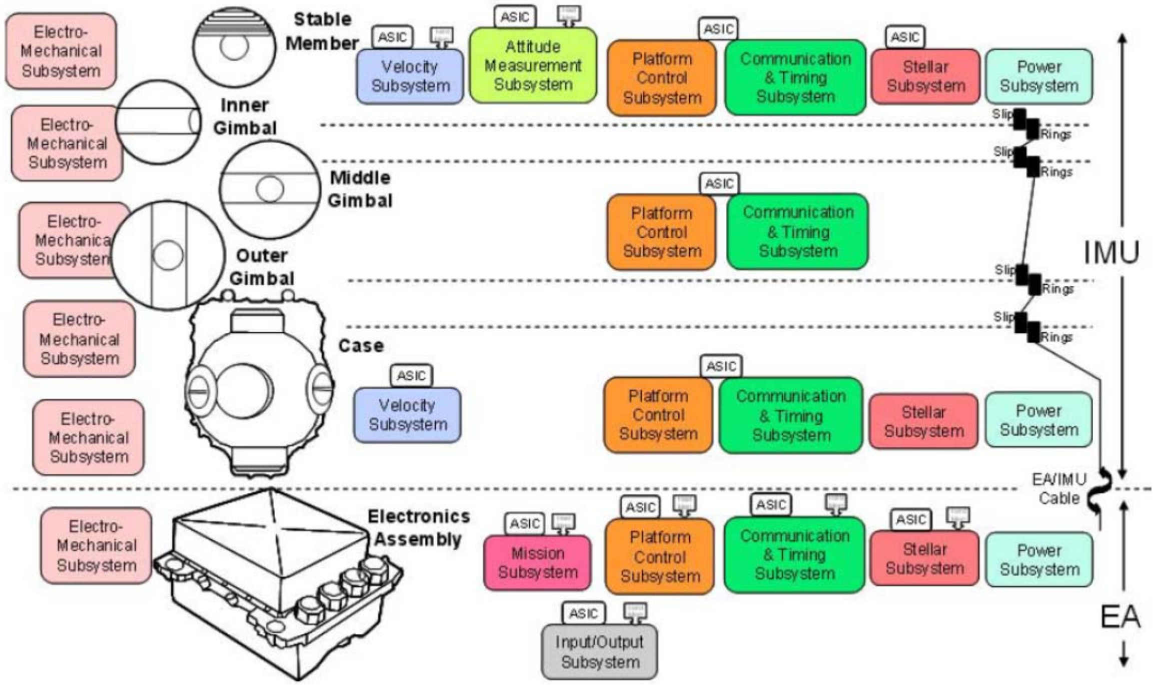
Electro-Mechanical Subsystem
Stable Member
ASIC
Velocity Subsystem
Attitude Measurement Subsystem
Platform Control Subsystem
Communication & Timing Subsystem
Stellar Subsystem
Power Subsystem
Electro-Mechanical Subsystem
Inner Gimbal
Slip Rings
Middle Gimbal
Electro-Mechanical Subsystem
Outer Gimbal
Platform Control Subsystem
Communication & Timing Subsystem
IMU
Electro-Mechanical Subsystem
Case
Velocity Subsystem
Platform Control Subsystem
Communication & Timing Subsystem
Stellar Subsystem
Power Subsystem
Electro-Mechanical Subsystem
EA/IMU Cable
Electro-Mechanical Subsystem
Electronics Assembly
Mission Subsystem
Platform Control Subsystem
Communication & Timing Subsystem
Stellar Subsystem
Power Subsystem
Input/Output Subsystem
EA

Figure 12. MK6LE Guidance System CONOPS. Source: Jackson et al. (2008).

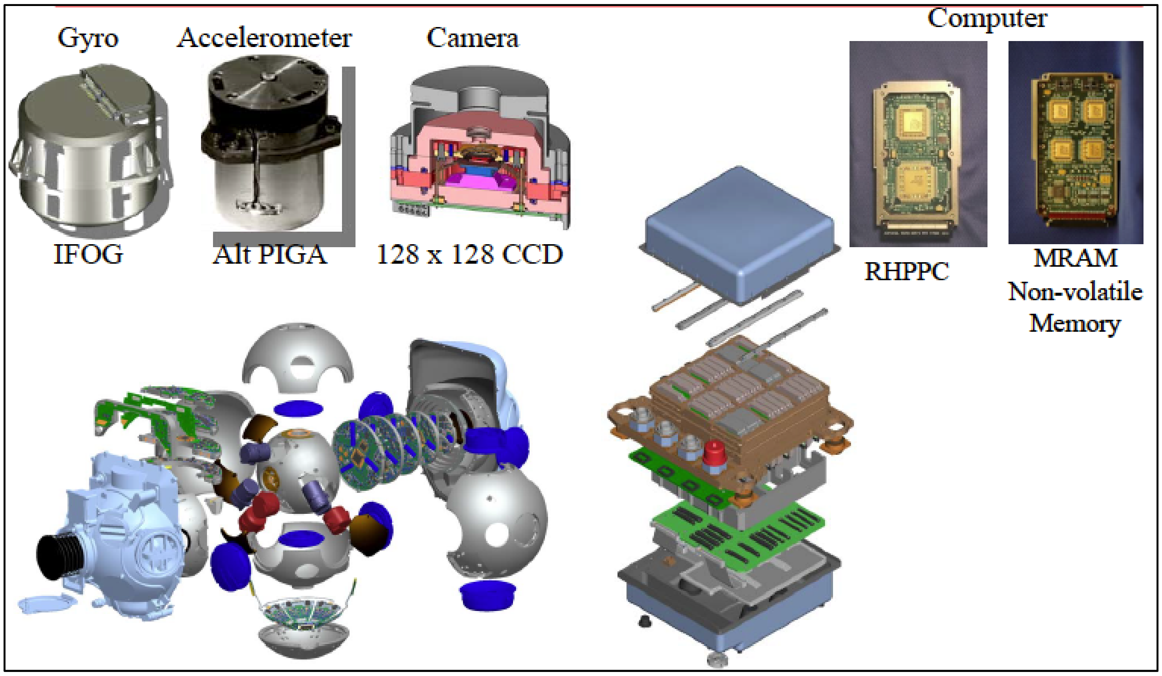

Figure 13. Physical Decomposition of MK6LE.
Source: Jackson, et al. (2008).

### 1. Software Development Process

Similarly to the previous two cases examined, the MK6LE project utilized a hybrid “spiral/Vee” process. Initial requirements, many of which were inherited from the legacy Trident program, were baselined before the design phase. As the SW implementation was matured, the SW was tested in the simulated environment. This allowed for a refinement of requirements and subsequent modification of the implementation in a spiral evolution toward the final configuration.

### 2. MBSE Approach

The MBSE process for the MK6LE started with a top-down architectural approach that evolved into a model-driven development activity. Using MathWorks’ Matlab and Simulink, a system-level model was created. From that system-level model, subsystem and component level models were implemented using a “black box” input/ output approach. As the teams matured their designs, the corresponding models for each subsystem were created, driven by the need to validate the detailed design before they were physically built.

When the MK6LE project started around the year 2003, MBSE and associated tools were not as mature as they are today. This condition led to the need for Draper to create many of the tools required to execute the MK6LE effort. The infrastructure and common support tools that were developed were referred to as the Virtual System Simulation model, or VSSim. As the central computing facility for the project, VSSim provided functions supporting early design activities such as auto-coding from interface control documents (ICDs). Nonetheless, the evolution from algorithmic model to developed C code for SW processing still required manual translation.

Enabling the capture of system standards, requirements, and design documentation, VSSim became the single information repository across the project life cycle. As the models were further decomposed, requiring more fidelity, a corresponding simulation layer would be introduced. Not only could the specific MK6LE system be matured and integrated early in the design stage, but also the effect of changes to other component systems could be modeled and quantified to understand the level of risk that could be expected from those external systems. The evolution of the simulation fidelity of the MK6LE shown in Figure 14 graphically demonstrates the subsequent granularity exercised across the development cycle.

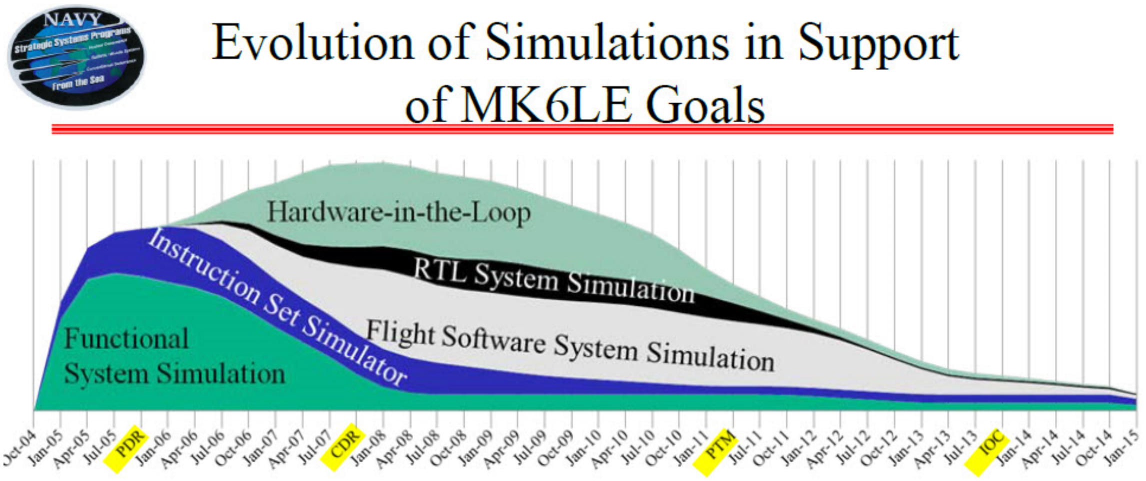

Figure 14. Simulation Fidelity of the MK6LE Across the Development Cycle. Source: Jackson, et al. (2008).

The initial plan and consideration given to the simulation needs, allowed the same core models to be used at every stage of simulation. This top-down model consideration for the required simulation allowed the MK6LE project to avoid the risk of having lower level model components not integrating together. The initial top-level model (functional system simulation) ran in a Simulink-based non-real-time environment, but with high order precision. Being able to create this model facilitated the next level of decomposition (the Instruction Set Simulator), which operated with SW as compiled for target HW. The instruction set simulator supported the synchronized simulations of four SW subsystems.

Since a large part of the MK6LE design involved the development of application-specific integrated circuits (ASICs) is a microchip designed for a special application, two of the critical modeling efforts involved register-transfer level (RTL) simulation. According to Semiconductor Engineering, RTL "is an abstraction for defining the digital portions of a design" (2016). Additionally,

> RTL is based on synchronous logic and contains three primary pieces namely, registers which hold state information, combinatorial logic which defines the next state inputs and clocks that control when the state changes. (Semiconductor Engineering 2016)

Once the models were capable of emulating the HW circuitry, the detailed ASIC design could be tested before any physical fabrication was initiated. This process was then taken one step further by interfacing the prototype system modules in a “hardware-in-the-loop” (HWIL) environment.

The MK6LE project’s MBSE process created SW models that supported the design needs at each stage of development. This approach allowed for initial missions to be flown in the lab, before any components were formally delivered. As the physical components were matured, the delivered HW would incrementally replace the SW simulation, eventually incorporating the actual flight processors and memory into the emulated system models. In this way, the MBSE process supported the end-to-end system development activities at a fraction of the cost that more traditional methods would have incurred.

# III. ANALYSIS

The following section details the comparison framework established to conduct the evaluation of the independent MBSE processes. The methodology for this effort considered which intrinsic factors of the MBSE approaches provided the best results to the performing organization. However, while a monetized value could be calculated for some factors, many of the driving attributes were deemed to be intangible, and did not lend themselves to a traditional cost-benefit analysis via direct quantitative measurement.

In order to assess the MBSE processes being studied, a trade-off evaluation approach was used. This approach was supported through the interactions with subject matter experts for the respective projects. Other evaluation techniques (rank order analysis, cost effectiveness/decision matrix analysis, etc.) were considered for this effort, but were discarded as they could be implicitly subjective, or would require a larger sampling of projects to be statistically meaningful. This work provides a framework that may be used to evaluate other programs or activities executing custom processes.

## A. ASSUMPTIONS

An initial assumption for this research was that there would be no significant differences in the life cycle models used in the respective acquisition and development efforts. It was also assumed that all organizations supported similar SE process standards. The data collection effort described in Chapter II validated these assumptions. All three projects involved large-scale, complex systems. They all utilized a modified Vee model, executing a Scrum software development process. All the projects also tailored existing tools to meet their development needs, and conducted model-based simulations at the lower (component) level.

## B. MBSE APPROACH EVALUATION

### 1. Evaluation Parameters

The criteria selected to support the comparison of the MBSE processes are listed in Table 2. These were largely based on the factors found in the open literature, particularly on the work by Sharon et al. (2010) as well as Alexander and Davis (1991), but were tailored to consider those parameters that would speak to the MBSE development processes and not specifically on the executing organization's structure or the programmatic details of the projects. The criteria were then used as a basis for analysis to be able to map and extract the key characteristics of each MBSE process.

Table 2. Final List of Criteria for Evaluation of MBSE Processes.

| Characteristic | Description |
|---|---|
| Requirements Maturity | The maturity of the customer's needs. |
| Scope Clearness | The clearness of the intent of the project, in terms of duration and deliverable products. |
| Stakeholder Commitment | The commitment of the stakeholders to pursue innovative approaches to system development. |
| Development Stability | The stability of the development environment. |
| Tool Availability | The extent to which MBSE tools were present and capable of performing the design synthesis functions. |
| Tool Supportability | The assistance vendors are able to provide to the end users concerning software programs and development tools. |
| Process Flow Definition | The identification of the MBSE process steps to allow the engineering team to transition through the phases in the development life cycle. |
| Ease of Implementation | The effort required to execute a task, given a set of tools. |

The final criteria were selected to help provide answers to the research questions representing the overall goals of this work. The first three criteria address the initial conditions for the development efforts. Requirements maturity and scope clearness provide an understanding of the volatility of the effort, as a change in direction can greatly affect a project particularly through the latter phases of development as well as impact the management of the requirements. Stakeholder commitment attempts to gauge whether the customer and other influential stakeholders are open to innovations as opposed to only performing tasks in a traditional manner, based on a review of the development processes and activities. Evaluated collectively, these three criteria provide a better understanding of how many of the requirements are absolute “needs” as opposed to “desires.”

The next four criteria (development stability, tool availability, and tool supportability and process flow definition) address the execution environment for the MBSE process, identifying the underlying suitability of the enterprise to conduct the development. These criteria provide an indication of a projects agility to adopt new development mechanisms, and for programs that do not use MBSE, indicate the potential to integrate those processes.

The final criterion (ease of implementation) while truly subjective provides valuable feedback when received from a subject matter expert. Insights are gained with respect to the level of resources and experienced personnel that may be required to adopt and engage in the respective MBSE process. Since one of the key goals of this thesis is to determine how to apply the MBSE approach to rapid acquisition programs, this was included as an important criterion.

### 2. Process Comparisons

The MBSE processes discussed previously were broken down to extract the main characteristics that define as well as differentiate them from each other. An initial view of the MBSE approaches identified numerous similarities. All the projects employed a top-down identification of system functionality using a modified Vee development process, combined with a Scrum approach to coding. All the efforts also tailored existing tools to meet development needs. However, it was the contrasting nuances and discriminating details between each process that are surmised to have provided considerable value-added in multiple ways. It is these traits on which this research focused on.

While all the projects created system models that were elaborately defined at a low (subsystem or component) level, the Trident program started at the higher system level and worked to implement models via a subsystem decomposition. This decomposition, which was a planned process activity, enabled the system to be architected from the top down, allowing requirements to be refined through the evolution of the subsystem models. As has been noted, this facilitated the re-use of core models throughout the development, allowing those models to be executed in every environment and system level.

All three projects conducted verification efforts through simulation, but the MK54 effort was able to take advantage of a large set (over 10,000 hours) of actual historical data by designing the models such that they could be stimulated by recorded at-sea data. The one key drawback presented by this approach was that there would be no "feedback" upon running the data into the model, as the recorded data captured historic events and could not be modified based on the "reactions" of the implemented models. However, this became a very powerful tool for verification, by allowing the models to be "run" through a wide range of test conditions, representing numerous environments, in a highly repeatable fashion.

Of the three MBSE approaches, the Radar program's process was deemed to have the most defined structure. Leveraging from mature documentation, work instructions, and commercial products, the Radar program employed a workflow that had become a common enterprise standard for the company. The workflow was documented by Finlay and Dujardin (2014) as having been similarly executed on a Naval Combat System program, as well as on an Integrated Air Missile Defense project, providing a consistent methodology to be able to execute the MBSE process across product lines. This level of reuse also provided cost savings to the respective customers.

### 3. Benefits of the MBSE Approach

The three projects evaluated used MBSE methods to specify, architect, implement, and verify their respective systems. In addition to supporting the capture of design information, the use of MBSE provided clear benefits to these programs. The three projects will be reviewed to highlight the advantages gained through their individual MBSE processes.

#### *a. MK54 Program*

The immediate benefit that could be identified with the Torpedo project was the impact to the testing. Prior to the MBSE-based re-architecture effort, system testing for code changes would require intensive engineering team support. As a result of the MBSE implementation, CSCI testing could be automated. The automation significantly reduced the amount of manual effort, along with the amount of time required to run specific test events, from 160 to five man-hours per release, reducing as well the need for a large team dedicated exclusively to supporting tests. In addition to providing greater repeatability and potential reduction in errors, the automation's reduction in manpower resulted in significant cost savings for the program.

By supporting a more modular design, the MBSE approach also allowed for testing of all the functionality in the CSCI. In the legacy implementation, only new functionality could be effectively tested as complete testing was prohibitive from a cost/ time standpoint. The automation not only allowed for thousands of tests to be performed (e.g., approximately 37,000 tests were run on the Classifier segment alone), increasing the overall test coverage.

The testing of the MK54 CSCIs also became more repeatable, with an improved accuracy (to 3 significant digits), providing better performance prediction prior to actual in-water runs, but also increasing the overall validation of the system design. These improvements, along with the "built-in" metrics for scoring the CSCIs discussed previously, had a massive impact to the Torpedo test program, enabling streamlined verification efforts with a large degree of agility and high accuracy.

Other benefits derived from the MBSE approach used on the Torpedo program were the ability to prototype and transition the technology to other efforts rapidly. Two cases where this capability was exercised were an effort associated with the Future Naval Capability for lightweight torpedoes, and the rapid prototyping development of a conceptual anti-torpedo-torpedo. These instances were discussed anecdotally, as the specific analysis of those efforts would require in-depth evaluation as was done for the

MK54 case. However, it can be stated that the reuse afforded by the MBSE methods resulted in additional costs savings.

### *b. Radar Program*

Through its graphical representation of system requirements, the Radar program produced high quality models. The visual representations provided a clear identification of the intended functionality being specified by the system designers and allowed for accurate refinement of requirements through increased stakeholder communication. The visualization of the system model helped to translate customer expectations instead of relying on seemingly endless amounts of traditional “shall” requirements.

Their MBSE process also reduced the manual documentation effort significantly by automating the creation of technical data packages and associated engineering materials, becoming an efficient turn-key operation. More significantly, through the establishment of a common MBSE approach, the process variability within the program was reduced, reducing the learning curve and associated late-stage defects. This common approach enabled the MBSE process to be efficiently transitioned from one program to another, as was also previously discussed. This transition across product lines was possible through the re-use of common components in individual customer models, which also decreased the timeframe to develop new customer-specific models, creating additional savings.

### *c. Trident Program*

The MK6LE project’s MBSE process provided the benefit of establishing a coordinated simulation environment that supported system development milestones throughout the entire design cycle. The modeling supported early system architecture trade studies and down-select, and was leveraged into the design integration activities through the verification and refinement of ICDs and early verification of subsystem requirements.

Similarly to the MK54 project, the MK6LE effort experienced significant benefits to their test program. The downstream validation and verification areas were greatly boosted through the use of the VSSim environment. VSSim enabled the capture of system standards, requirements, and early design documentation and became the single information repository across the life cycle by capturing the full design models and test data. By means of this approach in the Trident project's MBSE process, the modeling and simulation efforts could fully support the hardware and software development throughout the entire development effort.

By providing a robust simulation capability, developers could synthesize system components and validate them concurrently, facilitating the debugging of SW. As an example, for the HW, the developers were able to work on a model of the target processor before the actual HW was available. Through the virtual modeling environment, defects could be identified well before integration with physical HW ever took place, shortening the overall integration stroke. In this manner, custom hardware (ASIC) designs were proven to work the first time, out of initial production, without requiring either subsequent fabrication or additional non-recurring engineering efforts. The availability of functional subsystem models, and their evolution into functional operational components, were the principal factors that enabled the demonstrated execution of the MK6LE design by its preliminary design review, again at a reduced cost.

### 4. Key Enablers for the MBSE Approaches

A major objective of this research was to identify the elements of established MBSE processes that add the most value to their projects. These elements are characterized here as enablers, since their inclusion brought about significant performance improvements and cost savings to their respective programs. These enablers not only represent lessons learned from each MBSE process but are also seen as important factors which helped to make the three projects evaluated successful.

#### *a. Common Enablers*

Through the research to characterize the respective MBSE processes for each project, it became clear that the programs that had decided to undertake the use of MBSE had a customer who was committed to moving forward with this approach. While the formal concept of MBSE has been part of the systems engineering vernacular for almost

25 years (Wymore 1993), model-based practices have not yet become the norm for systems development, particularly in DOD programs. Due to this condition, there is some apprehension or even fear in the defense acquisition community toward MBSE development. As has been stated, implementation of MBSE can require additional effort during the initial development stages, constituting a certain level of "engineering inertia" from a perceived unrecoverable time and effort. Programs that are able to overcome this inertia, however, stand to gain the most from the benefits in the areas highlighted previously. The next few items identified herein are the enablers common to the three projects studied, contributing to their success.

Organizational Cachet—Part of the reason why the program customers were compelled to "buy into" the MBSE approaches was the organizational cachet brought about by the technical credentials that the individual performing organizations had demonstrated. In all cases, the developers had extensive relationships with the program offices. For example, the Torpedo developers had been the technical direction agent since the 1980s, Draper had been the design agent since program inception in the 1950s, and Raytheon could point to their track record from previous MBSE efforts. This organizational knowledge and prestige from the developers was translated into a confidence factor for their customers.

Stable High-Level Requirements—All three projects were the recipients of high-level requirements that went largely unchanged from inception through synthesis. This allowed the design teams to leverage initial conceptual evaluations and trade studies onto the architecture, preliminary and detail design phases. While designs were iterated in the development process, with stable requirements these iterations continually converged onto a final solution, without altering essential assumptions or implementation details.

Clearly-Defined Interfaces—All three projects took great care to study and identify their interface needs in terms of data fidelity and coverage. This early definition made the integration within the system models at each level easier but also reduced the risk that lower level components might not integrate together. In turn, the up-front planning and identification of interfaces also helped establish early "tap points" for simulation as well as data extraction.

### *b. Project-Unique Enablers*

In addition to the common factors highlighted previously, key enablers specific to each project were also identified. While these enablers would not be considered part of a standard MBSE approach, they integrated very well into their respective MBSE processes, and in all cases brought significant contributions.

Access to Historical Data—Of the three programs, the MK54 project was most directly able to leverage the vast archives of recorded at-sea data to support the MBSE development. By not completely discarding the infrastructure and data pipeline that had been established historically through 30 years of previous Torpedo efforts, the developers gained a trove of test vectors, with multi-faceted conditions across the spectrum of operating ranges upon which the final vehicle would be expected to operate. Not only did the use of historical data allow the designs to be tested "one step" removed from a live setting, but the use of recordings in an automated environment made the inputs repeatable to be able to validate new algorithms. For the MK54 project, this enabler impacted the development along both sides of the systems engineering Vee.

Tool Modification for Project Needs—For the Radar program, it was a seemingly simple change to the established Harmony process which became an enabler. Restructuring of the model browser allowed company-wide teams operating in different geographical locations concurrent access to the design. Through this subtle change, the program was able to draw in expertise from different areas when needed and integrate them into the development with minimal impact to personnel or physical resources.

Embedded Modeling and Simulation Team—The key enabler to the MK6LE project's MBSE process was the integration of the modeling and simulation team into the development team. Rather than being an independent entity or ancillary group, the Modeling and Simulation team was involved in every level of design. This utilization of the development team facilitated the reuse of core models at every stage of design, and enabled those models to effectively be turned into physical detailed designs.

## C. EVALUATION FOR INTEGRATION INTO THE SQQ89 PROGRAM

After completing the characterization of the MBSE approaches, an assessment of the target SQQ89 program was conducted through trade-off analysis and an evaluation of the pros and cons.

### 1. Implementation Factors for Consideration

Elements of these MBSE approaches could be adapted into rapid acquisition programs such as the SQQ89 by balancing the decision tradeoffs. As part of the process of defining the design space to meet primary project needs, conflicting performance factors would have to be evaluated within the constraints of the programs. For the SQQ89, the following list identifies the primary parameter tradeoffs that should be explored to address the key characteristics of that program:

- Project scale: With the inherent goal of bringing products to market at a fast pace, this element has to balance the desire to incorporate a large set of new capabilities against trying to execute a design within a compressed development timeline. As identified by Banner-Bacin (2009), the implementation of “MBSE can require additional effort during the development” (25) stages but has been shown here to have demonstrated benefits. These benefits, such as the reduction in design defects and increased capability to verify requirements, typically outweigh the extra upfront burden by reducing the overall design cycle time but has to be balanced in the context of the magnitude of a project. In other words, the amount of payback, or the time required to see tangible gains will be different for a small project versus a large program.

- Complexity: Part of the additional upfront burden in the MBSE processes comes about through the integration of modeling languages in what Banner-Bacin (2009) calls the “domain-specific applications, such as automotive, aerospace, communications, and information systems” (375). Model fidelity requirements also have to be carefully considered to support the specific model levels. For the case of the Trident system, trade studies were conducted to define the required timing constraints. For the functional system simulation, high-order precision SW models were created, albeit running at non–real-time. Conversely, for the HWIL, the time steps required to support the integrators were carefully selected to execute with the actual target HW (flight processor and memory) running fully in real-time.

- Evolution: When utilizing an incremental development paradigm, the rapid acquisition programs need to be mindful of deliberately evolving and growing a system capability against the potential sunk cost that could be lost from investing resources into a technology that remains immature. Additionally, as stated succinctly by Banner-Bacin (2009), “the use of MBSE can be difficult when upgrading legacy systems, subsystems or components due to the requirement to reverse engineer” (25).
- Traceability and Verification: In response to the need to evaluate the maturity of the product within the development phases, substantive validation and verification efforts are required. The use of an MBSE process facilitates this assessment, as was seen in particular with the MK54 program. Their MBSE approach allowed them to bake in metrics as part of the CSCIs so that implemented code could be evaluated by individual build, or by certain specific content within the CSCI.

### 2. Assessment of the SQQ89 Program

An evaluation of the SQQ89 target program was performed taking into consideration key characteristics of its development paradigm, many of which apply generically to other rapid acquisition programs. These can be summarized as follows:

- consists of a large scale project
- subject to an aggressive timeline to flow from design to production activities
- utilizes an incremental development paradigm
- re-using software, or planning to leverage off software re-use
- provides a solution to the classical detect/classify/localize problem
- requires validation/verification efforts to evaluate the maturity of the product within the development phases

These characteristics were then assessed in the context of the MBSE processes studied, through the use of the selected criteria and the implementation factors identified.

Requirements Maturity—As the SQQ89 incrementally adds capabilities, new system requirements are also incorporated. However, the top-level requirements, captured in the program’s operational requirements document, and verified in accordance with the test and evaluation master plan, do not change between ACBs. Thus, the high level mission for the program provides consistency, as was seen with the projects using MBSE.

Scope Clearness—At its most fundamental level, the SQQ89 provides an implementation of a “classical” detect/classify/localize problem. Furthermore, with defined acquisition process steps and expected system upgrades identified by fiscal year (e.g., ACB09 for 2009 and TI-11 for 2011) the duration of the development strokes and expected deliverable products are clearly projected out.

Stakeholder Commitment—The move to an APB-/ARCI–like process indicates the desire for the SQQ89 program to be agile and bring in new solutions. The SQQ89 program, however, requires robust validation and verification efforts to evaluate the maturity of the products, in order to support the tactical operational environment.

Development Stability, Tool Availability, and Tool Supportability and Process Flow Definition—The SQQ89 is subject to an aggressive timeline to flow from design to production activities, but the tools are stable and well known to the designers. However, the extent to which model-based engineering efforts are used is limited to the initial HW development process. For the SQQ89 HW, macro-level reliability block diagrams are developed using the Relex Studio modeling program to calculate the predicted system performance. A sample reliability block diagram is shown in Figure 15. The reliability block diagrams use a series of assumptions per MIL-HDBK-217 to generate results for failure rates that support the calculation of system Mean Time Between Failure (MTBF) and mean time between critical failure (MTBCF) system reliability rates.

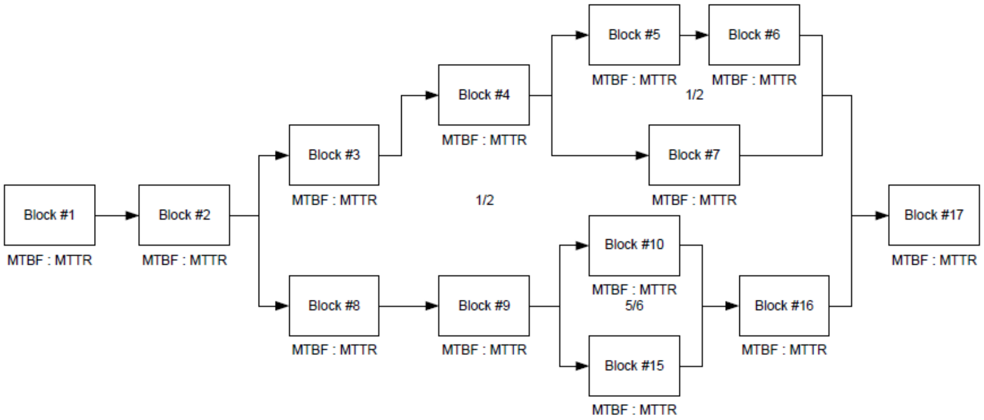

Figure 15. Sample Reliability Block Diagram.
Adapted from Lockheed-Martin, MST (2012).

Although the main focus of the modeling program is to define the hardware architecture, the reliability of the software architecture is also taken into account in this study. Since critical SW elements are required to have the highest reliability, the results of the HW allocations may impact where that SW. Although the instantiations from these models affect the overall system maintainability values, they fall short from being a system-wide model that can be used to verify other aspects of the system.

Ease of Implementation—Currently, the SQQ89 does not employ MBSE models. However, as part of its system supportability tools, the external inputs are well understood through synthetic training software. This understanding of what the external inputs are and how those interfaces are defined may enable the system architects to establish a clear demarcation of the SQQ89's boundaries so that a high-level model can be accurately instantiated.

Furthermore, the SQQ89 system is inherently defined by a set of functional segments. As an initial effort, it may be feasible to decompose the system level construct, and to build models for the functional segments being modified for a specific ACB variant. While this would require a gradual expansion of the models for each functional segment across ACBs, it may provide an evolutionary approach to supporting MBSE, as well as provide another means to evaluate the capability modifications within a specific ACB.

Since the ACB process is predicated upon an incremental construct, the incorporation of an MBSE approach could dovetail into the program’s development roadmap. The potential downstream impact to the verification and validation activities from applying MBSE would provide a clearer assessment of the maturity of the new capabilities for each ACB, to not only characterize the current as-is state, but also help define the desired to-be performance. The lack of existing MBSE processes in the SQQ89 may be the greatest obstacle for the program to overcome, but it is the author’s opinion that the payoff from incorporating these would be worth the effort.

# IV. SUMMARY

The primary goal of this research was to provide a framework upon which programs with aggressive timelines, in particular those classified as rapid acquisition software development projects, could compare themselves to in order to determine if they share some common traits with the "model" programs evaluated. This framework is summarized in the following steps: (a) the establishment of the evaluation parameters, (b) the comparison of the selected MBSE processes, (c) the identification of the principal implementation factors, and (d) the evaluation of the target program.

The research delineated the individual MBSE approaches of a set of defense acquisition programs and assessed the particular elements of their MBSE processes that served as differentiators to make them successful. A set of best practices and lessons learned from the programs that executed the MBSE approaches was provided to benefit those programs which have identified an intent or desire to implement MBSE into their programs as part of their software development efforts in the future.

After completing the characterization of the MBSE approaches, an assessment of the target SQQ89 program was conducted through a trade-off analysis based on a set of criteria established to evaluate the target program.

## A. CONCLUSIONS

This thesis sought to answer the specific questions identified in the introduction. These questions are re-stated and answers are briefly summarized as follows:

**What are the pros and cons of each of the processes?**

> In general, the main drawback was the initial effort required to establish an MBSE approach. The use of MBSE requires not only engineering rigor, but also programmatic commitment. However, once implemented, an MBSE design approach was invaluable for the three programs that were studied, as it supported each project's development goals. MBSE efforts focused not only on the early system requirements, design, analysis phases, but also the verification and validation activities throughout the later life cycle phases. MBSE allowed the programs to manage the evolution of simulation capabilities, as well as to assess the appropriate fidelity required to meet development needs.

**What elements of each process add the most value to their project?**

The three programs shared some common enablers that added the most value: (a) organizational cachet, (b) stable high-level requirements, and (c) clearly-defined interfaces. The organizational cachet gave the programs the confidence to embark with the using MBSE. While not restricted to MBSE programs alone, clarity of purpose, regarding the stability of requirements, interfaces, and approach were seen to contribute greatly to project success. Careful planning, supported by a holistic MBSE approach, brought about some project-unique enablers to each process. Whether leveraging their access to historical data, reuse of system design tools, or the embedding of the Modeling and Simulation team into the design effort, these intrinsic elements were used to remove developmental stovepipes. By exploiting all the capabilities of an MBSE approach, from design to validation, the programs were able to meet their development milestones successfully within the planned timelines.

**What attributes of the rapid acquisition projects might be improved from implementing an MBSE process?**

MBSE in general was identified as providing better quality requirements, resulting in lower rework. Combined with the gains achieved to the significant labor reduction due to automation, the MBSE approach provided improvements to the programs quality, schedule and cost. By providing repeatable test vectors with the required fidelity, the confidence levels associated with the designs was increased. The ability to automate testing and increase the test coverage allowed for a better assessment of model performance for existing functionality, as well as improving development and validation of new algorithms. Overall, the use of an MBSE approach helped improve the redesign, supportability and suitability of the programs reviewed.

Implementation of MBSE can require additional effort during the initial development stages but has demonstrated benefits, such as the reduction in design defects and increased capability to verify requirements, which typically outweigh the extra upfront burden by reducing the overall design cycle time. As presented through the discussion of the different case studies, the establishment of an MBSE approach requires programmatic commitment from the customer. There is an initial level of inertia and fear that needs to be overcome, but once commitment exists toward this investment, the payoff can be great.

By providing the ability to document designs and “capture and manage corporate intellectual property related to system architectures” (MBSE.Works 2015) the MBSE approach improves the traceability of requirements. Models developed at the system, subsystems, and CSCI levels could directly capture capabilities, functions, and requirements in a form that traced back directly to the customer needs. The visualization of the current approaches to MBSE is closing the gaps between the simulation and the actual systems themselves. As this coupling between the system models and application software becomes ever tighter, the line between the simulation and executable domains is being blurred to the point where in some cases a direct linkage can be created. In the words of Chris Finlay, who led the Radar MBSE effort, “the model *is* the design.” This direct instantiation may help boost the business case for promoting the use of MBSE, as customers see the corresponding value and impact from the use of models.

Rapid acquisition development demands high confidence. As was evidenced with the programs evaluated, designs could be quickly iterated so that they worked the first time out of the box. In fact, in all cases evaluated, MBSE resulted in an order of magnitude in time compression. As discussed with Dan Keating from Draper in a personal communication, “You use MBSE approach *because* you’re in a reduced cycle.”

Finally, the question was posed as to whether the adoption of MBSE, and incorporation of the enablers identified, could be introduced into the SQQ89 development approach or to other rapid acquisition programs to improve or optimize their development processes. The resulting evaluation identified the key characteristics of the projects using MBSE, and assessed them against the significant traits of the SQQ89 development efforts. Based on this framework, it is the author’s opinion that incorporation of MBSE processes would be recommended for the SQQ89. The scale, timelines, development paradigm, and verification needs of the SQQ89 program serve to justify the effort to engage in MBSE approach. As previously mentioned, the use of an MBSE approach provides palpable advantages from a technical, engineering standpoint. Nonetheless, it may be the programmatic benefits in terms of reduced time and cost which are needed to get customer buy-in of an MBSE approach. These advantages would certainly benefit current rapid acquisition programs, such as the SQQ89.

## B. RECOMMENDATIONS

This research starts to create a framework to identify the fit between the programs and the MBSE approach. An additional step that would be helpful toward this process would be to generate an in-depth mapping of the upfront MBSE work required. A detailed cost-benefit analysis would provide additional engineering rationale for embarking upon the use of MBSE. The follow-on assessment would serve to establish a business case to further justify the programmatic investment and overcome the fear of implementing an MBSE approach.

As an additional insight obtained through this work, it was evident that programs that applied MBSE at the lower levels, in particular the MK54 Torpedo program, expressed regrets of limiting the re-architecture to the CSCI levels of the system. From conversations with the MK54 subject matter experts who conducted a post-development introspective look, they stated that the execution level of the project was, in hindsight, too conservative. While the overall scope was kept very well defined, the program may have missed an opportunity to expand the benefits to system-level models which could have better quantified mission level requirements. This consideration should be weighed heavily by programs that are considering implementing MBSE as part of their SW development process.

## C. AREAS FOR FURTHER RESEARCH

Future considerations would be:

- to conduct a gap assessment of the available MBSE tools
- to introduce a Delphi-method assessment as a tool for surveying MBSE methods across the DOD industry segment, for a more comprehensive view of the penetration of MBSE approaches across the industry
- to integrate fully the available digital system models
- to leverage MBSE products further into the life cycle, for example to support system validation during field testing and perhaps supplement operational testing efforts

# LIST OF REFERENCES

Addington, Christopher D. 2008. “Model-Based Methodology for System of Systems Architecture Development with Application to the Recapitalization of the Future Towing and Salvage Platform.” Master’s thesis, Naval Postgraduate School.

Alexander, Linda C., and Alan M. Davis. 1991. “Criteria for Selecting Software Process Models.” Presented at the Computer Software and Applications Conference, 1991. COMPSAC’91 Proceedings of the Fifteenth Annual International, 521–528. IEEE.

Armstrong, Al. 2007. “AN/SQQ89 Overview.” Lecture, Naval Undersea Warfare Center, Newport, RI, September 27.

Bahlman, Jillian E. 2012. “A Model-Based Architecture Approach to Ship Design Linking Capability Needs to System Solutions.” Master’s thesis, Naval Postgraduate School.

Baker, Loyd, Paul Clemente, Bob Cohen, Larry Permenter, Byron Purves, and Pete Salmon. 2000. “Foundational Concepts for Model Driven System Design.” *INCOSE Model Driven System Design Interest Group, 16*. Jul. 15–16.

Banner-Bacin, Linda. 2009. “Application of Model Based Systems Engineering Methods to Development of Combat System Architectures.” Master’s thesis, Naval Postgraduate School.

Bjorkman, Eileen A., Shahram Sarkani, and Thomas A. Mazzuchi. 2013. “Using Model-Based Systems Engineering as a Framework for Improving Test and Evaluation Activities.” *Systems Engineering* 16(3): 346–362. doi:10.1002/sys.21241.

Bordonaro, Steve. 2012. “MK54 Torpedo Software Re-architecture: History and Lessons Learned.” Lecture, Johns Hopkins Applied Physics Laboratory, Laurel, MD, September 20.

Brock, David C., and Gordon E. Moore. 2006. *Understanding Moore’s Law: Four Decades of Innovation*. Philadelphia, PA: Chemical Heritage Foundation.

Buede, Dennis M. 2000. *The Engineering Design of Systems: Models and Methods*. New York: John Wiley & Sons, Inc.

Campbell, Andrew James. 2007. “An Evaluation of OMG SysML 1.0 a Standard Conformance between Modelling Tools.” Dissertation, Edith Cowan University.

Defense Acquisition University. 2013. “Chapter 10—“Decisions, Assessments, and Periodic Reporting.” *Defense Acquisition Guidebook.* Accessed July 31, 2016. https://acc.dau.mil/docs/dag_pdf/dag_ch10.pdf.

Demirci, Özlem. 2010. “Development of MBSE/UML Maturity Model.” Tekniska Jönköping School of Engineering, Master’s thesis.

Do, Quoc, Stephen Cook, and Matthew Lay. 2014. “An Investigation of MBSE Practices Across the Contractual Boundary.” Paper presented at the 2014 Conference on Systems Engineering Research, Redondo Beach, CA, March 21–22. Accessed March 18, 2016. doi:10.1016/j.procs.2014.03.083.

DOT&E. 2012. *Navy Programs, Mk 54 Lightweight Torpedo*. FY2012 Annual Report. Washington, DC: Director, Operational Test and Evaluation. http://www.dote.osd.mil/pub/reports/FY2012/pdf/navy/2012mk54.pdf.

Estefan, Jeff A. 2007. *Survey of Model-Based Systems Engineering (MBSE) Methodologies*. Rev B. Pasadena, CA: California Institute of Technology.

Forsberg, Kevin, and Harold Mooz. 1992. “The Relationship of System Engineering to the Project Cycle.” *Engineering Management Journal*, 4(3): 36–43.

Finlay, Christopher, and Stacy Dujardins. 2014. “From Theory to Reality: Taking the Fear Out of Model-Based Systems Engineering.” Paper presented at the 17th Annual Systems Engineering Conference of the National Defense Industrial Association, Springfield, Massachusetts, October 28–30, 2014.

Fitzgerald, Brian, Nancy Russo, and Tom O’Kane. 2000. “An Empirical Study of System Development Method Tailoring in Practice.” *ECIS 2000 Proceedings*, 4.

Frank, David, Kevin Hogan, and Shane Schonhoff. 2014. “Application of Model-Based Systems Engineering (MBSE) to Compare Legacy and Future Forces in Mine Warfare (MIW) Missions.” Master’s thesis, Naval Postgraduate School.

Giammarco, Kristin. 2010. “Formal Methods for Architecture Model Assessment in Systems Engineering.” Paper presented at the 8th Conference on Systems Engineering Research, March 17–19, 2010, Hoboken, NJ.

Global Security, 2011, “DD-963 SPRUANCE-Class.” Accessed July 29, 2016. http://www.globalsecurity.org/military/systems/ship/dd-963.htm.

Fuhrman, R. A. 1978. “The Fleet Ballistic Missile System: Polaris to Trident.” *Journal of Spacecraft and Rockets*, 15(5): 265–286.

Hoffmann, H. P. 2011. Systems Engineering Best Practices with the Rational Solution for Systems and Software Engineering, Deskbook Release 4.1. IBM Software Group. Accessed May 18, 2016. https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=dbc39547-3619-4c31-9535-0b583a4e6190#fullpageWidgetId=W62078615f88f_4809_afad_c27cdc9d7e71&file=2132d88d-4dde-40b4-8102-254ca4456c82.

Howie, Forrest, and Wayne O’Brien. 2012. “Enterprise Adoption of Model Based Systems Engineering.” Paper presented at the 2nd International Conference on Design and Modeling in Science, Education, and Technology: DeMset 2012, Orlando, FL, March 25–28, 2012. Accessed May 13, 2016. http://www.iiis.org/CDs2012/CD2012IMC/DEMSET_2012/PapersPdf/DC110RF.pdf.

INCOSE, 2007. *Systems Engineering Vision 2020*, version 2.03, INCOSE-TP-2004-004-02, Seattle, WA: International Council on Systems Engineering, Seattle. http://oldsite.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf.

Jackson, Todd, David Monti, Scott Berry, James Connelly, and Michael Murphy, M. 2008. “A Simulation-based Integration Approach for the First Trident MK6 Life Extension Guidance System.” Paper presented at the AIAA Missile Sciences Conference, Monterey, CA, November 18–20.

Johnson, Kipp M. 2010. “Tailoring Systems Engineering Processes for Rapid Space Acquisitions.” Master’s thesis, Naval Postgraduate School.

Johnson, William M. 2004. “The A-RCI Process—Leadership and Management Principles.” *Naval Engineers Journal* 116(4): 99–105.

Kaymal, Turgut. 2013. “Assessing the Operational Effectiveness of a Small Surface Combat Ship in an Anti-Surface Warfare Environment.” Master’s thesis, Naval Postgraduate School.

Letourneau, Jon P. 2009. “Incorporating Multi-Criteria Optimization and Uncertainty Analysis in the Model-Based Systems Engineering of an Autonomous Surface Craft.” Master’s thesis, Naval Postgraduate School.

Lockheed-Martin, MST. 2012. *TI-12 Reliability Block Diagram and Mathematical Models Report for the AN/SQQ-89 A(V)15 Surface Ships Undersea Warfare Combat System*. Doc. #77A128942, CDRL A080, Rev A, October 10. Manassas, VA: Lockheed-Martin Mission Systems and Training.

Lockheed-Martin, SSC. 2012. “Lockheed Martin-Built Trident II D5 Missile Achieves 137th Successful Test Flight.” Accessed April 27, 2016. http://www.lockheedmartin.com/us/news/press-releases/2012/march/lockheed-martin-built-trident-ii-d5-missile-achieves-137th-succe.html.

MacCalman, Alexander D. 2013. “Flexible Space-Filling Designs for Complex System Simulations.” Dissertation, Naval Postgraduate School.

MBSE.Works. 2015. “MBSE FAQ: Why Use MBSE?” Accessed December 26. http://mbse.works/mbse-faq/why-use-mbse.html.

Montgomery, Paul, Ron Carlson, and John Quartuccio. 2012. "System Definition-Enabled Acquisition (SDEA)—A Concept for Defining Requirements for Applying Model-Based Systems Engineering (MBSE) to the Acquisition of DOD Complex Systems." Excerpt from the Proceedings of the Ninth Annual Acquisition Research Symposium, Monterey, CA, May 16.

Naval Sea Systems Command PMS 4252. 1999. Acoustic Program Plan (Revision 8). Arlington, VA.

Naval-Technology. 2012. "US Navy Test Fires Trident II D5 Missile." Accessed April 27, 2016. http://www.naval-technology.com/projects/trident-ii-d5-fleet-ballistic-missile/.

———. 2016. "Trident II D5 Fleet Ballistic Missile, United States of America." Accessed April 27, 2016. http://www.naval-technology.com/projects/trident-ii-d5-fleet-ballistic-missile/.

Paredis, Chris. 2012. "Model-Based Systems Engineering: A Roadmap for Academic Research." Georgia Institute of Technology, Model-Based Systems Engineering Center. Accessed May 25, 2016. http://www.pslm.gatech.edu/events/frontiers2011/2.4_Paredis.pdf.

Program Executive Office—Integrated Warfare Systems 5. 2003. *APB Process Operating Instruction.* Washington, DC.

———. 2016. *AN/SQQ-89A(V)15 Systems Engineering Plan* (IWS 5–10-SEP-001). Washington, DC.

Pham, Phi, Keith Herndon, Doug Glandon, Mar Coble, Jeremy Royster, David Bailey, John Stewart, and Brandon Taylor. 2014. "Improving the Prototyping Process in Department of Defense Acquisition." Master's thesis, Naval Postgraduate School.

Pisani, Christopher R. 2013. "Linking Combat Systems Capabilities and Ship Design through Modeling and Computer Simulation." Master's thesis, Naval Postgraduate School.

Ramos, Ana L., José V. Ferreira, and Jaume Barceló. 2012. Model-Based Systems Engineering: An Emerging Approach for Modern Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42*, no. 1 (January 2012): 101–111. http://dx.doi.org/10.1109/TSMCC.2011.2106495.

Raytheon, 2015, "MK 54 Lightweight Torpedo." Accessed April 27, 2016. http://www.raytheon.com/capabilities/products/mk54/.

Robinson, Kevin, Despina Tramoundanis, David Harvey, M. Jones, and Shaun Wilson. 2010. "Demonstrating Model-Based Systems Engineering for Specifying Complex Capability." Paper presented at the Systems Engineering/Test & Evaluation Conference, Adelaide, May 3–6, 2010.

Rumsfeld, Donald, and Henry H. Shelton. 2001. Quadrennial Defense Review Report Department of Defense. Washington, DC, September 30, 2001.

Sánchez, Paul J. 2007. "Fundamentals of Simulation Modeling." In Proceedings of the 39th Conference on Winter Simulation: 40 years! The Best is Yet to Come, *IEEE Engineering Management Review* 37(2) 54–62. doi:10.1109/EMR.2016.2568698.

Sandhu, Reema. 2015. "Model-Based Software Engineering (MBSE) and Its Various Approaches and Challenges." *Compusoft* 4(6): 1841–1844.

Schmidt, Douglas C. 2006. "Model-Driven Engineering." *IEEE Computer Society*. Accessed May 25, 2016. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.106.9720&rep=rep1&type=pdf.

Schwaber, Ken. 2004. *Agile Project Management with Scrum*. Redmond, WA: Microsoft Press.

Semiconductor Engineering. 2016. "Register Transfer Level: An Abstraction for Defining the Digital Portions of a Design." Accessed July 29, 2016. http://semiengineering.com/kc/knowledge_center/Register-Transfer-Level/49.

Sharon, Itamar, Michel dos Santos Soares, Joseph Barjis, Jan van den Berg, and Jos LM Vrancken. 2010. "A Decision Framework for Selecting a Suitable Software Development Process." Master's thesis, Delft University of Technology.

Steward, Victoria. 2015. "Functional Flow and Event-Driven Methods for Predicting System Performance." Master's thesis, Naval Postgraduate School.

Tepper, Nadia A. 2010. "Exploring the Use of Model-Based Systems Engineering (MBSE) to Develop Systems Architectures in Naval Ship Design." Master's thesis, Massachusetts Institute of Technology.

Topper, J. Stephen, and Nathaniel C. Horner. 2013. "Model-Based Systems Engineering in Support of Complex Systems Development." *Johns Hopkins APL Technical Digest*, 32(1): 419–432.

United States Navy. 2013. "United States Navy Fact File, MK 54—TORPEDO." Accessed April 27, 2016. http://www.navy.mil/navydata/fact_display.asp?cid=2100&tid=1100&ct=2

Walker, Stephen K. 2005. Capabilities-Based Planning-how It Is Intended to Work and Challenges to its Successful Implementation. U.S. Army War College Strategy Research Project, Carlisle, PA.

Wilson, Anthony M. 2009. “The Impact of Software Reuse on the Cost of Navy Sonar and Fire Control Systems.” Master’s thesis, Naval Postgraduate School.

Wymore, A. Wayne. 1993. *Model-Based Systems Engineering*, Boca Raton, FL: CRC Press.

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California