NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

SYSTEMS ENGINEERING CAPSTONE REPORT

COMMON COMBAT SYSTEMS—REMOTE SOFTWARE DEPLOYMENT OF INCREMENTAL CAPABILITY AND UPDATES

by

Daniel W. Carter, Cedric T. Chiu, Timothy O. Hillman,
Roianthony Navarro, Harsh Shah, and Victoria Tsugawa

December 2020

Advisor: Brigitte T. Kwinn

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2020	**3. REPORT TYPE AND DATES COVERED** Systems Engineering Capstone Report

4. TITLE AND SUBTITLE
COMMON COMBAT SYSTEMS—REMOTE SOFTWARE DEPLOYMENT OF INCREMENTAL CAPABILITY AND UPDATES

5. FUNDING NUMBERS

6. AUTHOR(S) Daniel W. Carter, Cedric T. Chiu, Timothy O. Hillman, Roianthony Navarro, Harsh Shah, and Victoria Tsugawa

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
NSWC PHD, Port Hueneme, CA 93043

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.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

U.S. Navy ships need to have regular software updates to ensure their cyber security posture is sound and to ensure their sailors have what they need to defend the ship against any attack. Naval Surface Warfare Center Port Hueneme Division (NSWC PHD) wants to find a process to rapidly deploy the software from a centralized location to a dispersed fleet. Currently software is hand carried by a team of personnel to perform the software upgrades, which takes a lot of time and money to execute.

This project proposed to use a remote deployment system (RDS) process and investigated the feasibility of using model-based systems engineering (MBSE). A system architecture was created using Cameo. The team researched methods that industry currently uses to see what can be adapted for use with the U.S. Navy ships. This project scenario investigated software upgrades to the Aegis platform while the ship is in port for the proof in concept.

Monte Carlo simulations and analysis confirmed the feasibility of transferring large files over low bandwidth scenarios. A decision model was developed to determine which alternative held the most value. Analysis showed that the RDS would be much more beneficial to the command as opposed to the current way of conducting business.

14. SUBJECT TERMS remote software deployment, remote software upgrade, integrated software upgrade, software upgrade	**15. NUMBER OF PAGES** 87
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

COMMON COMBAT SYSTEMS—REMOTE SOFTWARE DEPLOYMENT OF INCREMENTAL CAPABILITY AND UPDATES

Daniel W. Carter, Cedric T. Chiu, Timothy O. Hillman,

Roianthony Navarro, Harsh Shah, and Victoria Tsugawa

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2020

Lead Editor: Timothy O. Hillman

Reviewed by:
Brigitte T. Kwinn
Advisor

Accepted by:
Ronald E. Giachetti
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

U.S. Navy ships need to have regular software updates to ensure their cyber security posture is sound and to ensure their sailors have what they need to defend the ship against any attack. Naval Surface Warfare Center Port Hueneme Division (NSWC PHD) wants to find a process to rapidly deploy the software from a centralized location to a dispersed fleet. Currently software is hand carried by a team of personnel to perform the software upgrades, which takes a lot of time and money to execute.

This project proposed to use a remote deployment system (RDS) process and investigated the feasibility of using model-based systems engineering (MBSE). A system architecture was created using Cameo. The team researched methods that industry currently uses to see what can be adapted for use with the U.S. Navy ships. This project scenario investigated software upgrades to the Aegis platform while the ship is in port for the proof in concept.

Monte Carlo simulations and analysis confirmed the feasibility of transferring large files over low bandwidth scenarios. A decision model was developed to determine which alternative held the most value. Analysis showed that the RDS would be much more beneficial to the command as opposed to the current way of conducting business.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION....1
A. PROBLEM OVERVIEW....2
B. PROJECT OBJECTIVES....3
C. PROJECT SCOPE....3
D. CURRENT PROCESS....3
E. CHAPTER SUMMARY....5

II. SYSTEMS ENGINEERING PROCESS....7
A. INITIAL RESEARCH PHASE....7
B. PROBLEM REFINEMENT PHASE....8
C. ANALYSIS OF ALTERNATIVES PHASE....8
D. FINAL PROCESS DESIGN....8
E. TECHNICAL APPROACH....9
F. SYSTEM LIFE CYCLE....10
G. CHAPTER SUMMARY....13

III. RESEARCH....15
A. LITERATURE REVIEW....15
B. IMPORTANT DEFINITIONS....15
1. Compile to Combat in 24 Hours....16
2. Consolidated Afloat Network Enterprise and Services....16
3. Risk Management Framework....17
4. Integrated Battle Command System....18
5. Operation and Maintenance Center....18
6. Operational Readiness Test System Tech Assist Remote Support....19
7. Supervisory Control and Data Acquisition....19
C. CHAPTER SUMMARY....20

IV. PROBLEM REFINEMENT....21
A. NEEDS ANALYSIS....21
B. CONOPS....22
C. GAP ANALYSIS....23
1. Current System....24
2. Proposed System: Remote Deployment System....24
D. REQUIREMENTS ANALYSIS....25
1. System Requirements....25

2. Stakeholder Requirements. 25
3. Functional Requirements 25
4. Non-functional Requirements 27
E. CHAPTER SUMMARY 30

V. SYSTEM ARCHITECTURE 31
A. METHODOLOGY 31
1. System Models 31
2. State Machine Diagram 32
3. System Boundary Model 34
4. Use Case 34
5. Action Diagrams 35
B. FUNCTIONAL ANALYSIS 36
C. CHAPTER SUMMARY 36

VI. ANALYSIS OF ALTERNATIVES 37
A. POTENTIAL SOLUTIONS 37
1. Satellite (in port) 37
2. Proximity Radio Frequency Link (in port) 39
B. FEASIBILITY ANALYSIS 41
C. MONTE CARLO SIMULATION 44
1. Design of the Monte Carlo Simulation for the RDS 44
2. Simulation Results 44
D. CHAPTER SUMMARY 46

VII. RESULTS AND CONCLUSIONS 47
A. RESULTS 47
1. RDS Model 47
2. Monte Carlo Simulation 48
B. RECOMMENDATIONS 48
1. Recommendations Based on the Current Work 48
2. Recommendations for Future Work 48

APPENDIX A. CONFIGURATION MANAGEMENT PLAN 49
A. PURPOSE 49
B. OBJECTIVE 49
C. APPROACH 50
D. ORGANIZATION 50
E. TRAINING 50

APPENDIX B. RISK ANALYSIS 51
A. THE RISK MANAGEMENT PROCESS 51
B. RISK IDENTIFICATION 54
1. Risk 1—Technical Maturity of RDS 54
2. Risk 2—Stability of Connection and Correctness of Software Downloads 55
3. Risk 3—Network Security 55
4. Risk 4—Training 55
5. Risk 5—Integration Issues 56
C. RISK ANALYSIS 56

APPENDIX C. MONTE CARLO SIMULATION SETUP 57

LIST OF REFERENCES 59

INITIAL DISTRIBUTION LIST 63

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. High-Level Diagram Showing the Transfer of Software Upgrades from the Developers to the Shipboard Systems 1

Figure 2. Current Process for Delivering Software Updates to the Fleet 4

Figure 3. Tailored SE Process. Adapted from Buede (2000). 7

Figure 4. System-of-Interest for the RDS 11

Figure 5. Software Life Cycle Processes. Adapted from ISO/IEC/IEEE 12207 (2017). 12

Figure 6. CONOPS Graphic 22

Figure 7. CONOPS SysML Diagram 23

Figure 8. Functional Requirements in SysML 26

Figure 9. Non-functional Requirements in SysML 27

Figure 10. Performance Diagram 28

Figure 11. RDS System Models Including Behavior and Structure Diagrams 32

Figure 12. State Machine Diagram for the Software Update Process 33

Figure 13. System Boundary Model 34

Figure 14. UC Diagram of the RDS 35

Figure 15. Functional Block Diagram Hierarchy 36

Figure 16. BDD for Satellite Alternative 38

Figure 17. IBD for Satellite Alternative 39

Figure 18. BDD for Proximity Radio Alternative 40

Figure 19. IBD for Radio Alternative 41

Figure 20. Value Hierarchy 42

Figure 21. Monte Carlo Simulation Results for 500 Runs 45

Figure 22. Monte Carlo Results Time Bar Chart 45

Figure 23. Monte Carlo Results Download Times....46

Figure 24. DOD Risk Management Process. Adapted from Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics (2006). 52

Figure 25. Risk Reporting Matrix 56

Figure 26. Monte Carlo Action Diagram 58

LIST OF TABLES

Table 1. Decision Model for Alternative System Selection.....43

Table 2. Parameters Used in Monte Carlo Setup.....44

Table 3. Levels of Risk Occurrence. Source: Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics (2006).....53

Table 4. Levels of Risk Consequences. Source: Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics (2006).....54

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ACS	afloat core services
AoA	analysis of alternatives
ATO	authority to operate
BDD	block definition diagram
C2C24	compile to combat in 24 hours
C4ISR	command, control, communications, computers, intelligence, surveillance, and reconnaissance
CANES	consolidated afloat network enterprise and services
CCE	common computing environment
CD	compact disk
CEC	Cooperative Engagement Capability
CI	configuration item
CM	configuration management
CMP	configuration management plan
CNO	Chief of Naval Operations
CONOPS	concept of operations
COTS	commercial-off-the-shelf
CS	combat system
CSSQT	combat system ship qualification trials
DISA	Defense Information Systems Agency
DOD	Department of Defense
DODI	Department of Defense Instruction
DoN	Department of the Navy
DTIC	Defense Technical Information Center
EIA	Electronic Industries Alliance
EW	electronic warfare
FD/FI	fault detection and fault isolation
GB	gigabytes

GUI	graphical user interface
HMI	human machine interface
IAVA	information assurance vulnerability alert
IBCS	integrated battle command system
IBD	internal block diagram
IEEE	Institute of Electrical and Electronic Engineers
IS	information systems
IT-21	Information Technology for the 21st Century
LAN	local area network
MB	Megabytes
Mbps	Megabits per second
MBSE	model-based systems engineering
MIL-STD	military standard
MOE	measure of effectiveness
NAVSEA	Naval Sea Systems Command
NAVWARSYSCOM	Naval Information Warfare Systems Command
NCES	net-centric enterprise services
NPS	Naval Postgraduate School
NSWC PHD	Naval Surface Warfare Center – Port Hueneme Division
OMC	operation and maintenance center
OPNAVINST	office of the chief of naval operations instructions
ORTSTARS	operational readiness test system technical assist remote support
PIT	Platform Information Technology
PLC	programmable logic controller
PO	program office
RDS	remote deployment system
RF	radio frequency
RMF	risk management framework

RMMCO	Regional Maintenance and Modernization Coordination Office
SATCOM	satellite communication
SCADA	supervisory control and data acquisition
SCD	ship change document
SE	systems engineering
Sea Power 21	Sea Power in the 21st Century
SEP	systems engineering plan
SID	ship installation drawing
SIPRNet	secret internet protocol router network
SOA	service oriented architecture
SRA	selected restricted availability
STIG	security technical implementation guide
SySML	system modeling language
TCP	Transmission Control Protocol
TI	technical instruction
UC	use case
UDP	User Datagram Protocol
UML	Unified Modeling Language
V&V	verification and validation
WSERB	Weapon Systems Explosive Safety Review Board

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The Department of the Navy (DoN) has increasing requirements to update combat systems (CS) onboard various Navy vessels (Moore 2019). The current process to deliver and update the CS software can take weeks depending on ship availability. The objective of this project is to develop a process that supports the incremental software upgrades of CS on Aegis destroyers and the ensuing certification of the CS with the new software that is faster than today's process. The intent of the new process is to provide a delivery, installation, and certification time frame of two to three days, instead of weeks with the current system.

The objective is to develop a process that supports common CS incremental capability software updates that is faster than the current process and obtain the necessary evidence required to certify the CS. The team will use model-based systems engineering (MBSE) to develop the process, develop a configuration management plan (CMP), and provide a concept of operations (CONOPS) diagram. Figure 1 shows the high-level diagram recommended for the delivering CS software onboard ship.

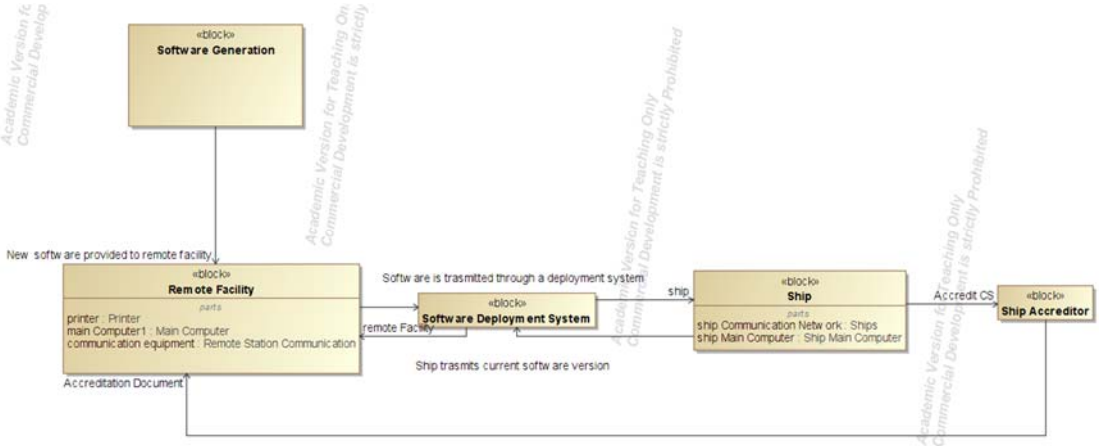

Figure 1. Recommended Process to Deliver CS Software

This paper captures the systems engineering (SE) approach to outline a new streamlined process to rapidly deploy CS software and obtain the necessary evidence required for certification or software version information for the CS. There are many different approaches and methods that the industry uses to push software updates to various

devices. This paper researched the industry methods and provided feasibility analysis if it can be used in the Navy. Overall, using a SE approach and MBSE methods, the recommendation is a new process based on feasibility analysis.

The team followed the Vee model SE process adapted from Buede (2000) and the software life cycle technical processes adapted from ISO/IEC/IEEE 12207:2017. This project only covered the left side of the Vee model because the objective was to determine feasibility of the remote deployment system (RDS), not to realize the system. Literature research conducted revolved around a remote software deployment system for both commercial and military applications. The team developed the CONOPS of how the RDS would deliver CS software packages and maintain configuration management (CM). From the CONOPS, the team developed functional and non-functional requirements that the RDS would need to satisfy to push software updates and query ships to retrieve current software version numbers.

Using Cameo Enterprise and Innoslate as the prime software tools for the System Modeling Language (SysML) models, the team followed the modeling methodology adapted from the *Practical Guide to SysML* by Friedenthal, Moore, and Steiner (2015). The team created SysML models to describe the interface between external entities and internal subsystems, as well as the how the system behaves based on the CONOPS. The SysML models provided a graphical representation of how the RDS will function in an operational environment, as well as a means to link defined subsystems to their respective requirements.

Two alternatives for the RDS were analyzed against the current method of physically delivering and updated CS software. A decision model, including a value model were created. Based upon the analysis, the current system has higher reliability since technical personnel are physically onboard installing the data. However, the RDS reduced the number of personnel required and the time required to deliver and install the software, demonstrating that the RDS the more viable alternative.

The team used Innoslate to conduct Monte Carlo simulation for the RDS transmitting software packages of various sizes and various download speeds. The software

package sizes and download speeds were chosen based on common satellite speeds. From the simulation analysis, 68% of the software will be transmitted within one hour and 19 minutes for the recommended system. Further analysis will need to be conducted for a defined range of software packages and bandwidth to acquire a refined estimate of transmission time.

Based on the literature research conducted, the technology exists to establish communication between a remote facility and the ship's network. The team recommends further analysis of systems that provide a means to deploy software from a remote location.

References

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

Friendenthal, Sanford, Alan Moore, and Rick Steiner. 2015. *A Practical Guide to SysML: The Sytems Modeling Language.* Waltham, MA: Elsevier Inc.

ISO/IEC/IEEE 12207:2017(E) First Edition 2017–11: ISO/IEC/IEEE International Standard - Systems and Software Engineering -- Software Life Cycle Processes. 2017. IEEE.

ISO/IEC/IEEE 15288 First Edition 2015–05-15: ISO/IEC/IEEE International Standard - Systems and Software Engineering -- System Life Cycle Processes. 2015. IEEE.

Moore, Vice Adm Thomas J. 2019. *Technology Development Roadmap - Naval Power and Energy Systems.* Roadmap, NAVSEA, Arlington: U.S. Navy, 45.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Naval Surface Warfare Center Port Hueneme Division (NSWC PHD) needs a process to deliver new software capability to ship CS and support the CS to get certified for use. This process will include a RDS to aid in carrying out the process. There is no process at NSWC PHD right now that remotely delivers software upgrade capability to the CS and support to all the ships. This project will help NSWC PHD standardize a new, efficient process of supporting shipboard CS faster than they currently do.

Figure 1 shows at a high level what the process of pushing CS incremental capability and software updates to Navy vessels currently looks like. The software updates are generated by developers and then reviewed by stakeholders. Future stakeholders include the program office (PO) sponsor, NSWC PHD, and certification authority. Currently, there are no stakeholders for this project. Once stakeholder approval is given, the files are then sent to a remote location where the file can be sent to the ship to be installed.

Figure 1. High-Level Diagram Showing the Transfer of Software Upgrades from the Developers to the Shipboard Systems

A. PROBLEM OVERVIEW

The solution needs to include a process for how shipboard systems will be accredited through the risk management framework (RMF) authority to operate (ATO) process, and certified through other agencies such as the Weapon Systems Explosive Safety Review Board (WSERB). Currently, when shipboard CS get upgraded, there are several approvals and processes that must be completed under Military Standard (MIL-STD)-882 Series, Office of the Chief of Naval Operations Instructions (OPNAVINST) 8020.14 Series, and Naval Sea Systems Command (NAVSEA) instruction 8020.6 Series. Ship change documents (SCDs) need to be approved, and part of that process is getting approval from WSERB. Other guidelines for the SCD process are outlined in technical instructional (TI) guidance document TS9090-400A. Approval from the planning yards for the ship installation drawings (SIDs) is also necessary, where applicable. The next step is to be put on the schedule during the ship's selected restricted availability (SRA) which is scheduled by the Chief of Naval Operations (CNO) (Aftergood 2020). After the upgrade has been installed and verified on the ship by the engineers and technicians, typically there are combat system ship qualification trials (CSSQTs) to show CS requirements have been met. These CSSQTs are scheduled at-sea events that qualify the entire CS. Land based infrastructure also needs to be taken into consideration because once the CS is pushed to the ships, support will be needed in case there are issues with the installation, maintaining interoperability, and compatibility with existing systems (SEA06L 2019).

The primary stakeholders are NSWC PHD, the software configuration and certification organizations, and the NAVSEA POs for the ships and the CS. Ship's force will receive new software upgrades and support through a RDS. NSWC PHD will be in charge of the process to deliver the software capability and the necessary support to certify the CS. NSWC PHD will also provide the subject matter experts that provide insight to what CS needs to be upgraded and how often the upgrades need to occur, and the project managers responsible for the ship's schedule. The POs will be responsible for providing funding.

B. PROJECT OBJECTIVES

The project objective is to develop a process that supports common CS incremental capability software updates that is faster than today's process and obtain the necessary evidence for CS certifications. The team will use MBSE to develop the process, develop a CMP, and provide a CONOPS diagram.

C. PROJECT SCOPE

The scope of this project will be focused on the development of a process to rapidly deploy software updates from a centralized location such as the NSWC PHD to a dispersed fleet. This process will include delivering software updates to shipboard CS, automatically tracking the installation of the software updates, verifying afloat software configurations, and providing necessary objective quality evidence to support RMF certifications. Processes external to this project will be the development of new system and software capabilities, the functions of the POs, and the activities of non-Navy government organizations and support contractors.

For this project, the team used the Aegis Baseline 10 as the common CS. For this baseline, there are two types of software updates. The first, major baseline updates, are accomplished prior to deployment. The second type of software updates are monthly, quarterly, or bi-annual updates for small fixes such as addressing information assurance vulnerability alerts (IAVAs). These more frequent software updates are dictated by the Defense Information Systems Agency (DISA), and can occur as often as every 30 days. For this project, we examined the bi-annual, major baseline updates. The scenario for addressing the DISA updates will be explored by a future cohort.

D. CURRENT PROCESS

The current process for distributing major baseline updates is based on the policy established in IEEE/EIA Standard 12207, and is shown in the flow chart depicting the process in Figure 2 where the steps taken to update three ships with new software will be examined. In this scenario there are a limited number of installation teams available which results in a situation where the fleet's CS have to be updated in sequential order.

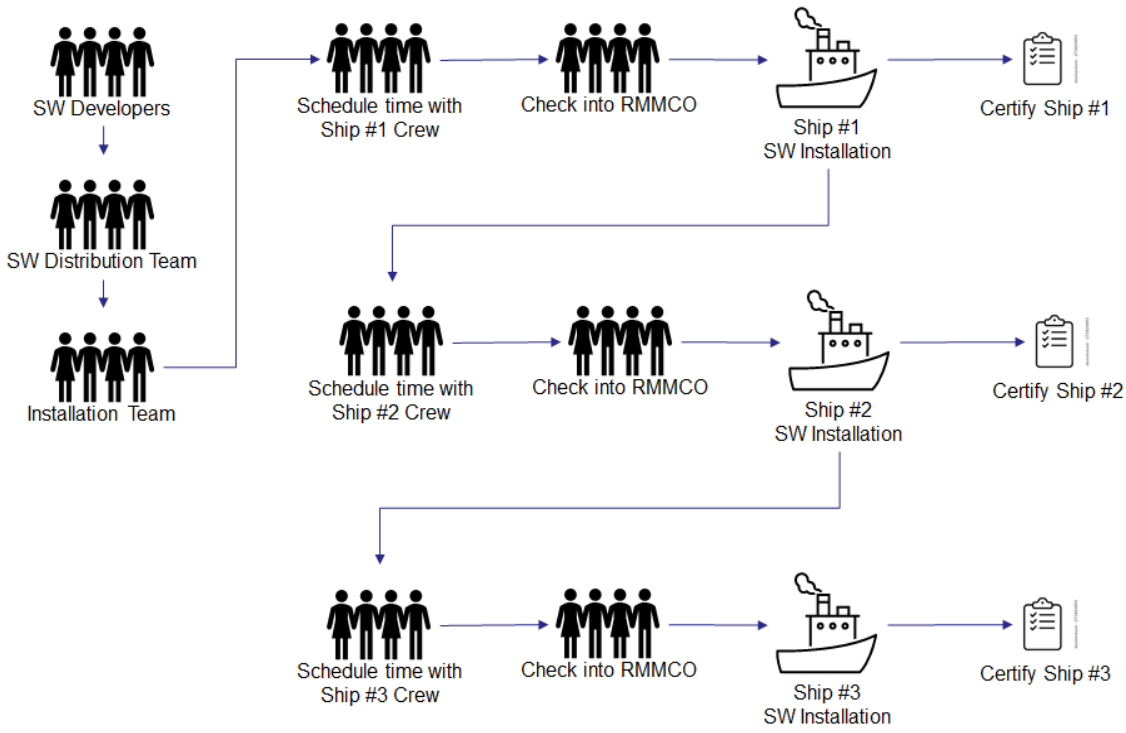

Figure 2. Current Process for Delivering Software Updates to the Fleet

In the first step in the current software update process, the software developers send the final software updates to the distribution team. The distribution team then sends the software to the installation team. The installation team then needs to schedule time with every ship that needs the software update. The installation team will reach out and schedule a time with the crew from Ship #1 for when they are in port, and when they have available time for both the software installation and the testing to ensure the system is working properly. Once that is scheduled, the installation team will travel to where the ship is, and check in with the Regional Maintenance and Modernization Coordination Office (RMMCO). This step is mandatory prior to starting any software installations to ensure all of the approvals and paperwork are in order. After that they can go to the ship and perform a pre-test to check the current CS is operational, install the incremental software update, and then perform a test after the installation to ensure the CS works correctly with the new software update. After that the system needs to be scanned by the installation team to show the system is secure. Upon completion of a successful update, the scans are sent off to the

accreditation team to certify Ship #1. This process is then repeated for every other ship that requires the software update, and typically takes several weeks of planning and execution. It results in a time-consuming process that takes weeks for the update to be installed and certified for every ship in the fleet with the same baseline configuration.

E. CHAPTER SUMMARY

This chapter examined the current process to accomplish software updates on CS aboard ships in the fleet. The analysis looked at the roles of the stakeholders and their involvement in the process. The current process was explained in order to identify deficiencies that the new process will attempt to improve or eliminate. The scope of the project that will design a new system that will be more efficient in use of time and manpower resources while still providing a secure and manageable process was also outlined.

THIS PAGE INTENTIONALLY LEFT BLANK

II. SYSTEMS ENGINEERING PROCESS

The final product for this capstone is a process, so the solution focused on the left side of the Vee model. For this project there were four phases. First, the initial research phase, then the problem refinement phase, then the analysis of alternatives (AoA) phase, which led to the last phase, the final process design. These phases and Vee model are shown in Figure 3.

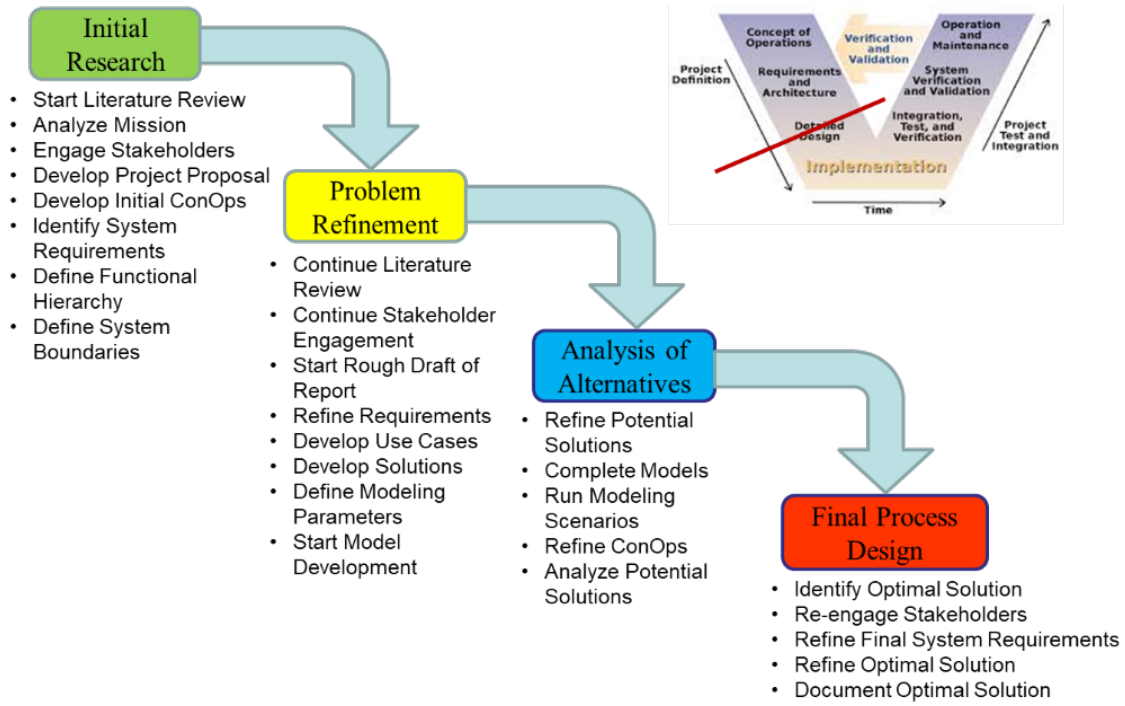

Figure 3. Tailored SE Process. Adapted from Buede (2000).

A. INITIAL RESEARCH PHASE

The initial research phase started with a literature review, which helped with the problem definition. It provided insight on how other industries and services are deploying software updates with minimal human intervention. The vision owner was consulted to clarify the project description, scope, and objective.

During the initial research phase, the initial system CONOPS, requirements, functional hierarchy, and system boundaries were defined. This was done during the initial research phase to ensure the scope of the project was defined, and to give a clear direction on what is and what is not covered by this capstone. These were refined in the later phases of the SE process.

B. PROBLEM REFINEMENT PHASE

The second phase of the SE process is the problem refinement phase. This is where a continuation of the literature review and vision owner engagement took place, and requirements were refined. Models were developed to show use cases (UCs), parameters, and the beginnings of verification and validation (V&V). The system concept definition defined functional requirements and functional architecture. Tools such as the Cameo Enterprise Architecture were used to create SysML models and diagrams.

C. ANALYSIS OF ALTERNATIVES PHASE

During the AoA phase, the CONOPS and mission scenarios were finalized and used to model the solutions developed in the problem refinement phase. Both the solutions being modeled, and the modeling parameters were modified to show a thorough analysis of the proposed solutions. Multiple scenarios were simulated, and the recommended solution was identified based upon the results of the modeling compared with the measures of effectiveness (MOEs) and the system requirements.

D. FINAL PROCESS DESIGN

The last phase of the SE process for this project was the final process design phase. This is where the recommended solution was identified. The vision owner was re-engaged to ensure the solution fits the need for NSWC PHD. After comments from the vision owner were adjudicated, the final system requirements and optimal solution were refined. Once everything is acceptable to the vision owner, the recommended solution was documented.

E. TECHNICAL APPROACH

The technical approach explains the reasoning of the methodology, technical path and tools used. The technical approach is in the order of problem definition, stakeholder analysis, AoA, analysis of interface requirements, system concept definition, architecture design, model integrations with external system, simulation with UCs, and risk mitigation strategy

The technical approach started with creating a problem definition, completing a literature review, and reaching a system requirements definition. This portion utilized tools such as Google Scholar, Dudley Knox Library, and other electronic means for literature search.

In the stakeholder analysis, operational details, requirements, mission profiles, and MOEs were defined using feedback from subject matter experts and the vision leader. The high-level operational graphic and CONOPS models were generated using Cameo and SysML.

The AoA was based on capability and gap analysis, and feasibility assessment of two alternatives. The first alternative is to keep the current process, and the second alternative is to remotely provide the updates to the ship while the ship is in port. Each alternative was measured with parameters such as bandwidth constraints, the number of personnel required to execute the delivery, and the number of days to plan and execute the delivery.

The architecture design used tools such as Cameo, Innoslate or other visual Unified Modeling Language (UML) applications. During the architecture design, detailed block definition diagrams (BDD), internal block diagrams (IBD), UC diagrams, and sequence, activity and state diagrams were generated.

The risk mitigation strategy described in Department of Defense Instruction (DoDI) 5000.02: Operation of the Defense Acquisition System and Department of Defense (DOD) acquisition risk management guidelines was used for risk impact analysis, risk matrix generation, and various analyses of risk.

F. SYSTEM LIFE CYCLE

This paper covers the design portion of the system life cycle as shown in Figure 3. Tailored from the left side of the Vee model, the process will include stakeholder needs and requirements definition, a CONOPS of the new process, modeling of different processes, and V&V of the new process.

The process created will be used in the operational stage of a CS and ship life cycle. The system of interest is the system that delivers the CS software to the ship. In accordance with ISO/IEC/IEEE 24748-1:2018, the CS software deployment system will continue to support the ship's life cycle until the ship is decommissioned and goes into the retirement portion of the life cycle (2018). The software deployment system interacts with the ship in the operational environment with the software generation system, the records management system, and the accreditation system. Figure 4 is the system-of-interest diagram adopted from ISO/IEC/IEEE 24748-1:2018.

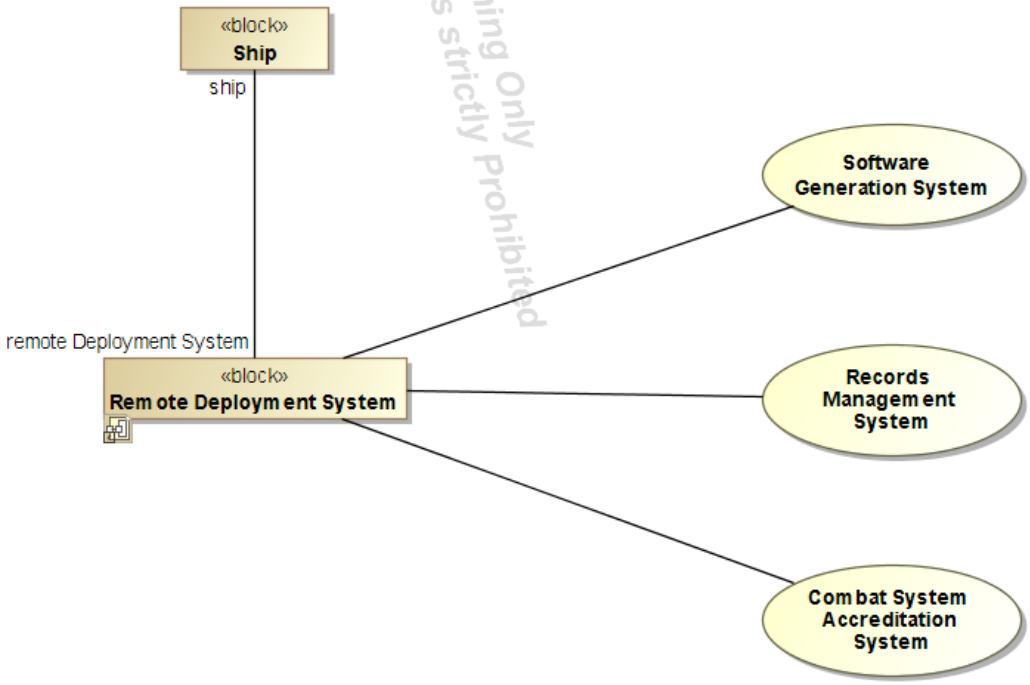

Figure 4. System-of-Interest for the RDS

From the software perspective, ISO/IEC/IEEE 12207:2017 contains processes that are specific to software. This standard is adopted and aligned to the ISO/IEC/IEEE 15288:2015, which describes the system life cycle processes not specific to software but systems in general. Figure 5 shows the software life cycle processes from ISO/IEC/IEEE 12207:2017. This paper will cover some of the technical processes and technical management processes for the RDS. Future work will be required outside of this paper to complete the remaining portions of the technical processes.

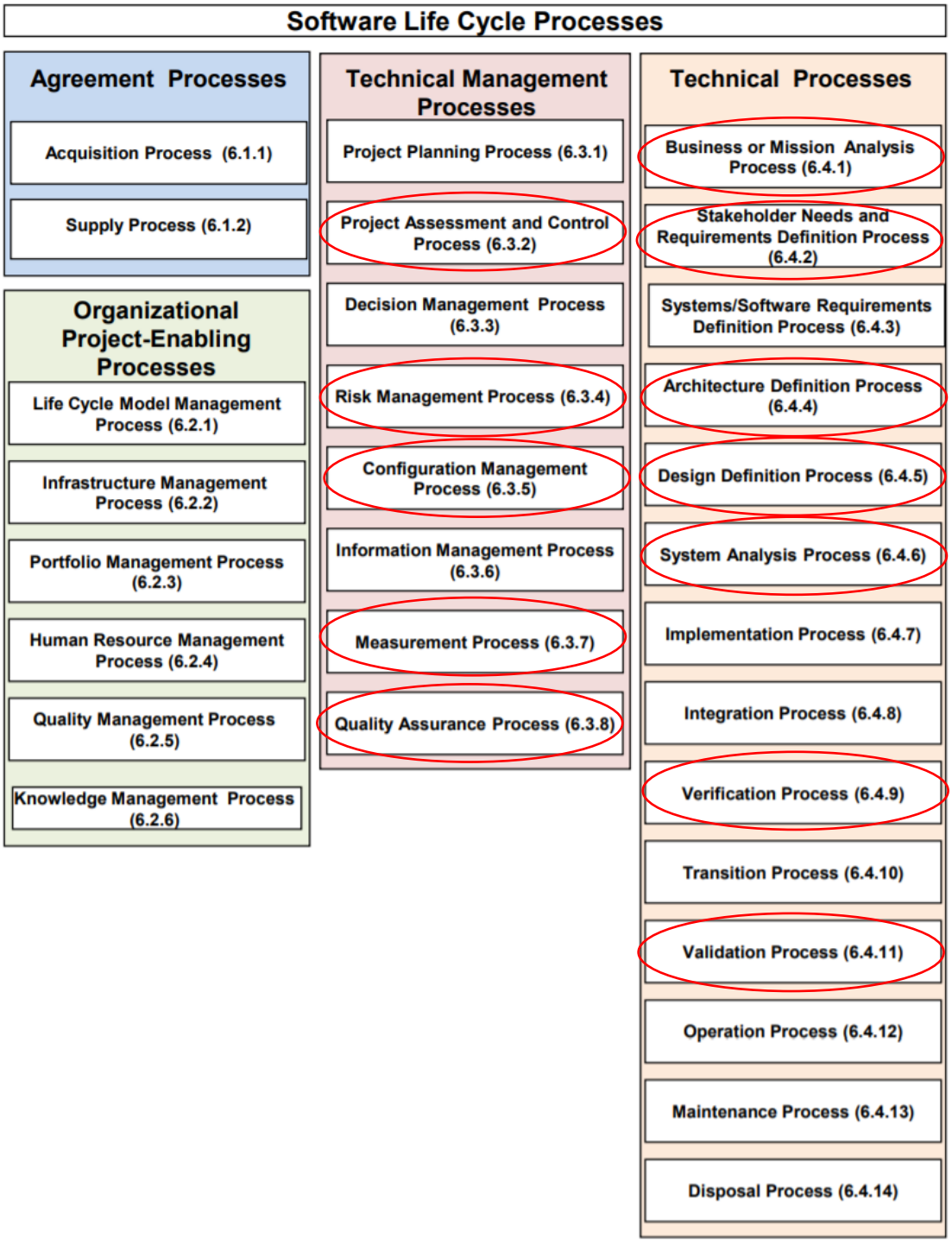

Figure 5. Software Life Cycle Processes. Adapted from ISO/IEC/IEEE 12207 (2017).

G. CHAPTER SUMMARY

This chapter discussed the Vee model, tailored from Buede 2000. The project will focus on the left side of the Vee model, from problem definition to conceptual system design. The system life cycle and technical approach that was used to develop a process that supports common CS incremental capability software upgrades was also described.

THIS PAGE INTENTIONALLY LEFT BLANK

III. RESEARCH

This chapter describes the research related to the project and solution development, and the scope of the literature review and the findings related to remote software deployment, data transfer, various networks, and certification and accreditation processes.

A. LITERATURE REVIEW

A literature review was conducted to scope the problem and identify potential solutions. The types of literature the team reviewed include academic theses, research papers, conference proceedings, Navy doctrine, and other DOD doctrine. In total, the team reviewed 15 documents and selected nine to comprise the literature review. The team used Google Scholar, Defense Technical Information Center (DTIC), and the Naval Postgraduate School (NPS) Dudley Knox Library to search for applicable literature.

Research was conducted in the following general areas:

- networks currently on the ship
- remote software upgrade capabilities
- RMF
- communication networks used by the commercial sector to deal with similar issues

This research helped to scope and bound the problem and potentially identify some solutions and measures.

B. IMPORTANT DEFINITIONS

In order to understand the relationships of the systems and the boundaries, the following terms were defined and will be used in this capstone project.

1. Compile to Combat in 24 Hours

The Compile to Combat in 24 Hours (C2C24) program was developed by Naval Information Warfare Systems Command (NAVWARSYSCOM) in 2018. “The concept is based on web services, or micro-services, similar to those you would see on your smart phone, and use of a new cloud architecture Navy is developing and testing, and fielding this capability quickly and securely.” (*CHIPS* 2018)

The architecture consisted of four key pillars:

- Data Standardization
- Use of shared infrastructure
- Automating functional and cybersecurity controls testing
- Use of the Cloud (*CHIPS* 2018)

The pilot test was executed with the existing Consolidated Afloat Network Enterprise and Services (CANES) shipboard infrastructure on numerous Navy vessels. “The pilot demonstrated that the Navy does not need to test, certify and install a new piece of hardware every time it wants to deploy a new software capability.” (*CHIPS* 2018) The difference between C2C24 and this project is NAVWARSYSCOM does not work on CS upgrades, so it is not applicable for the NSWC PHD need.

2. Consolidated Afloat Network Enterprise and Services

“The CANES program is being developed to address the issues of the legacy shipboard network situation.” (Rognlie 2010, 8) The stated goals of the CANES program are:

- “build a secure afloat network required for Naval and Joint operations”
- “consolidate and reduce the number of afloat networks through the use of mature cross-domain technologies and Common Computing Environment (CCE) infrastructure”

- "reduce the infrastructure footprint and associated costs for hardware afloat"
- "provide increased reliability, application hosting, and other capabilities to meet current and projected Warfighter requirements"
- "Federated Net-Centric Enterprise Service (NCES) Afloat Core Services (ACS) to the tactical edge to support overall DOD Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) applications migration to a Service Oriented Architecture (SOA) environment" (Rognlie 2010, 8)

3. Risk Management Framework

"The DOD RMF describes the DOD process for identifying, implementing, assessing, and managing cybersecurity capabilities and services, expressed as security controls, and authorizing the operation of Information Systems (IS) and Platform Information Technology (PIT) systems" (AcqNotes 2020). One of the main objectives of the RMF process is to incorporate cybersecurity into the development process as early as possible and to enforce that cybersecurity is addressed throughout the project stages. For systems that are accredited, there are mandatory monthly scans for the systems to ensure there is timely corrections of the vulnerabilities.

When the DOD has a new IA system, or a system that has had an upgrade, the system must be scanned for cyber vulnerabilities per the RMF process. Before the system is scanned for the cyber vulnerabilities, system owners must implement a predetermined list of Security Technical Implementation Guides (STIGs) to the system. An example of a STIG is to ensure there is a DOD warning banner when the system is first turned on with an acknowledge button for the user to click. Once the STIGs are implemented, the system is scanned with a software tool to ensure the STIGs have been met. If the STIG cannot be met because it will hinder the system from performing its job, the system owner can write a statement explaining why the STIG cannot be met. The scans and explanations are sent to RMF accreditors, and they review the findings. If there are issues, it goes back to the

system owners for adjudication. Every time a new software load is installed to the system, the system will need to be scanned and re-certified. It is anticipated that major software updates will be installed prior to each deployment and shall receive bi-annual updates for minor software changes.

4. Integrated Battle Command System

Integrated Battle Command System (IBCS) is a networked and integrated system of sensors, shooters, and mission command like the U.S. Navy's Cooperative Engagement Capability (CEC). The IBCS program provides one solution to the DOD's requirement for an adaptable joint fire control and sensor platform. The IBCS has the capability to provide updates in a multi-domain environment. The IBCS system does not transfer software updates but is a system of integrated systems that share information through secure channels. This network of systems is a set of communication lines between multiple radars, mission commands, and launch platforms so that multiple shooters can see beyond line of sight, all track numerous targets, and determine which shooter is optimal for engagement. This system communication has framework established that could serve as an outline for this effort.

5. Operation and Maintenance Center

The Operation and Maintenance Center (OMC) is defined as the specified location that uses minicomputers or personal computers to monitor multiple terminals and transmits data to different equipment terminals through a communication network (Dai 2010). Along with equipment terminals, the OMC is an embedded system that monitors and remotely pushes software updates to equipment when necessary. This process utilizes flags and upgrade states to indicate whether a packet is successfully delivered. In this system, however, the packet will still be available in the network, which will be something the team will need to consider when design this system since the team's data may contain sensitive information.

6. Operational Readiness Test System Tech Assist Remote Support

Operational Readiness Test System Tech Assist Remote Support (ORTSTARS) is a shore-to-ship, remote access system to the Aegis CS. The system utilizes a network-centric communication structure based on the Information Technology for the 21st Century (IT-21) tactical shipboard local area network (LAN), as well as the Secret Internet Protocol Router Network (SIPRNet). This is a digital collaborative environment. After organizing a connection time via SIPRNet, sailors on board ships can communicate with on-shore personnel and discuss troubleshooting the system. Fault detection and fault isolation (FD/FI) data undergo a bi-directional exchange in real time. The two main benefits of ORSTARS is the ability to distance support, and a method for rapid tech assist on demand. The ORTSTARS concept is based on the Navy's Sea Power in the 21st Century (Sea Power 21) strategic plan for building a surge-capable force. The hardware architecture is based on the "open-system architecture" and leverages commercial-off-the-shelf (COTS) equipment.

All current U.S. Navy Aegis cruisers and destroyers have the capability of conducting distance support via the ORTSTARS system. The ORTSTARS physical connection is based on the SIPRNet IT-21 LAN and Ethernet Category 5 cables.

7. Supervisory Control and Data Acquisition

The Supervisory Control and Data Acquisition (SCADA) system "incorporates programmable logic controllers (PLCs), human machine interface (HMI) workstations and network communication systems into a complete integrated system" (McCrady 2013, 1). By utilizing user-defined data values, logic operations can be programmed into the system. This allows programmers to set flags for any variable of the stakeholder's interest. The SCADA system utilizes these logic controllers in conjunction with HMI workstations, and the network allows users to communicate with the system from an onsite workstation or a remote site assuming the remote site has access to the network. In previous projects, SCADA has been used to monitor the status of power plant operations.

McCrady provides very detailed explanation of the SCADA software in his book with architecture examples showing how the SCADA system operates and emphasizing

the importance of documentation for software systems. For this project, the elements of the SCADA system will serve as a good reference to the type of system the team will incorporate into current existing systems.

C. CHAPTER SUMMARY

The literature review was conducted in order to scope the problem of this project and to help identify potential solutions. The research was conducted on networks currently onboard ships, remote software upgrade capabilities, RMF, and other projects or companies that are achieving similar goals and overcoming similar issues. This research helped us identify potential solutions, measures and ways to model the system.

IV. PROBLEM REFINEMENT

This chapter will document problem refinement. First, needs analysis is discussed, then an overview of the CONOPs, the gap analysis which will cover the current system and the proposed system, and finally the requirements analysis.

A. NEEDS ANALYSIS

Technology is evolving at a very fast pace, and there are increasing requirements to update CS onboard various Navy vessels (Moore 2019). If there were a way to push updated CS software or programs to the vessels at a rapid pace, numerous vessels could be updated in a very short amount of time. NSWC PHD needs to have a process to deliver new software capability to ship CS and support the certification of the CS for use after the software installation.

Due to the aggressive schedules and urgency to update CS software when needed, a new streamlined process to push updates to the ship will be required. After a new software package has been developed, there is a need to reduce the amount of deploy the software to the ship to obtain necessary CS certifications. Additionally, there is a need quickly verify the current afloat configurations installed on board. From these two main needs, a streamline process can be generated which will reduce the overall update and certification time frame throughout the fleet.

The current software upgrade delivery process requires the installation team to schedule a tentative timeline with the ship crew to hand deliver and install software. The software installation team waits for ships to return to port. During this time, the team travels to the port, checks in with the port's RMMCO, and goes onboard with the physical media to perform the update. The proposed process will eliminate the sending personnel to install the media with the ability to remotely deploy or update software upon return to port and a monitoring system showing the software version for each ship.

B. CONOPS

The remote deployment process for updating CS software is intended to speed up the process of delivering adaptation data to the ship and receive CS certifications. After the software development team finishes the software package, the software is sent to a remote ground facility. The remote ground facility is intended to be a station where the software can be pushed to their respective ships without an installation team coming onboard. Instead, ship's force in charge of CS will be notified of the software push, following procedures to acquire the software package and update CS. After software installation, the ship conducts tests and evaluations to acquire certifications from an approval authority. Certifications are reported back to the remote ground facility, which can be accessed at any point. A graphical view of the operational concept is show in Figure 6.

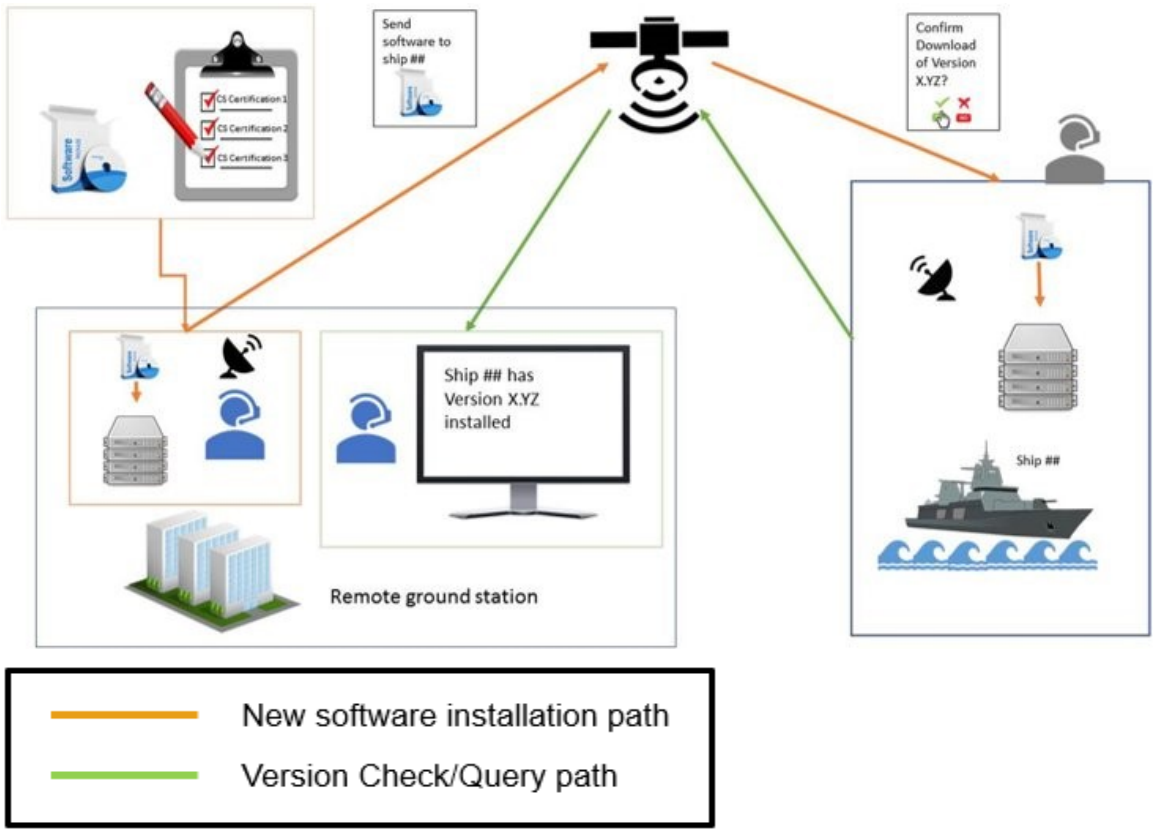

Figure 6. CONOPS Graphic

Figure 7 shows the SysML structure diagram for the CONOPS. The software development team is shown in the software generation block. The software will then be transferred to the remote facility, which contains a main computer which stores the software and communication equipment. The communication equipment will communicate with the hardware of the software deployment system, linking the ship's communication system and the remote facility. After link is established, the software is then transmitted to the ship into the ship's main computer where the software can be installed into the CS. The CS is assumed to be within the ship block.

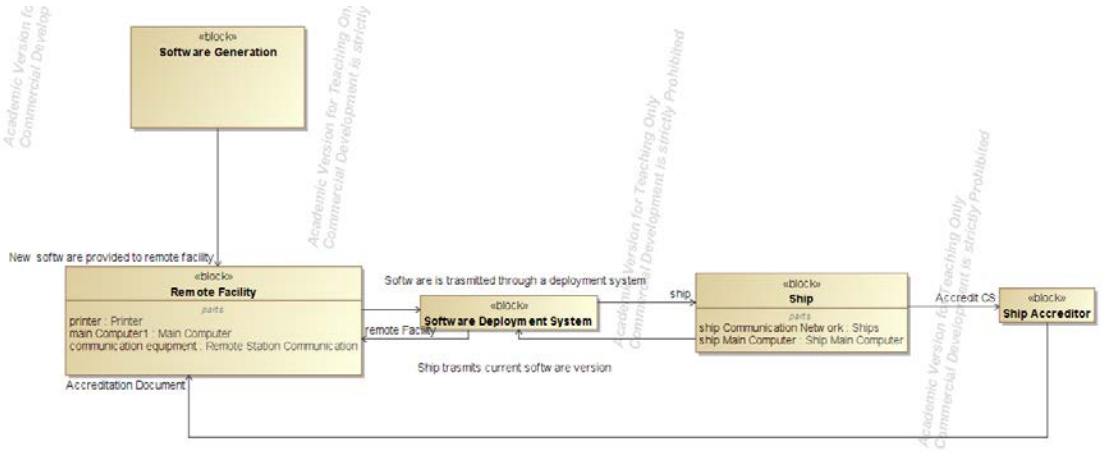

Figure 7. CONOPS SysML Diagram

C. GAP ANALYSIS

The software delivery system under analysis for use by the NSWC PHD is designed to be a means to get software updates to ships while away from port. The updates will be sent from a land-based station at NSWC PHD via remote communications to the intended ship or fleet of ships. The reason for this development is to keep on-board CS software current and protected from vulnerability in potentially hostile environments. Initially the objective of this system is to update and maintain Aegis destroyers with basic or minor software updates and successfully maintain fleet software compliance while in deployed status.

1. Current System

Currently NSWC PHD has no remote software update capability. The only method that is available for NSWC PHD to be able to push updates to Aegis destroyers is to perform updates while the ship is in port. This action is performed by physically carrying a compact disk (CD) on-board and loading the update directly to the system. Alternately, the same action can be performed while the ship is at sea but requires a person to be transported to the ship while at sea. The first scenario only happens every 12–18 months when the ship(s) pull into port. This poses a significant risk to the ships because of the vulnerability of the ship to cyber threats (and other means of electronic warfare (EW)) when in potentially hostile regions or situations. The second scenario can occur at a reduced interval time, but still comes with significant risk. Depending on the location of the ship at time of need for update, the risk of transportation of personnel (civilian or military) can be extremely high. Both of these situations carry significant risk to life and ship, but also carry a high logistics burden. The cost of performing either of these scenarios is high, especially considering the infrastructure is already in-place to be able to perform the remote deployment.

2. Proposed System: Remote Deployment System

The proposed system for NSWC PHD to update software for CS on deployed ships would eliminate the need for the direct touch aspect to the update; or at least removing the requirement for someone to travel to the physical ship. In this process, the update would take place by remotely pushing the update from a central location at NSWC PHD. This update would be most comparable to the update that a cell phone receives from a cellular provider. The update would be developed and vetted similar to the current process, but in this process the hand-over to the ship would be performed via satellite or through some type of untethered communication method. The update would be accepted and completed by an appointed person on the ship. The primary objective for this software update system is to be able to provide critical software updates to Aegis CS that could be vital to system performance during stressing times. There are, of course, other objectives for having the

ability to be able to push updates more frequently and easily, but for because of project constraints only system performance is considered.

D. REQUIREMENTS ANALYSIS

This section describes the different requirements for the RDS. This section describes the requirements in order to meet the need to efficiently deliver CS software to the ships.

1. System Requirements

System requirements were based upon the vision owner's interpretation of the requirements. The literature review and research also ensured we considered sets of requirements (Buede 2000). The requirements were partitioned from the stakeholder requirements to include functional, non-functional, constraint and interface requirements.

2. Stakeholder Requirements.

The stakeholder requirements are to deliver CS software to the ship after new software data is required. The CS software shall successfully install the correct software into the CS. The CS software shall be validated and verified that the data in the software is correct. The version of the CS software shall be tracked and recorded for traceability.

3. Functional Requirements

This section of the report will outline and itemize the "shall" aspects of the proposed system. The foundation of the functional requirements is a mixture of stakeholder needs and vision owner subject matter expertise. These requirements are a top-level description of the CS software and its interaction with the Aegis destroyer.

- The system shall be designed for updating CS software on an Aegis destroyer.
- The system shall be designed to support multiple future CS elements that would require software updates to reduce total ownership cost.

- The CS elements shall receive major software updates prior to deployment, and shall receive bi-annual updates for minor updates.
- The ship systems shall receive updates when ship is not under distress/any type of battle mode.
- The boundary of the designed system shall not include the certifying organizations.

The functional requirements model in SysML is shown in Figure 8. The top block represents the overall requirement to deploy CS updates to the ship. This requirement is refined to five child requirements shown in the next level. These requirements include a subsystem that needs to deliver the updates, a subsystem to receive the updates, a subsystem that tracks configuration of the CS software, and a subsystem that maintains records and documentation of accreditation. The lower tiers shown in the SysML diagram refine the child requirements to further define when CS software will be delivered and how software updates and tracked.

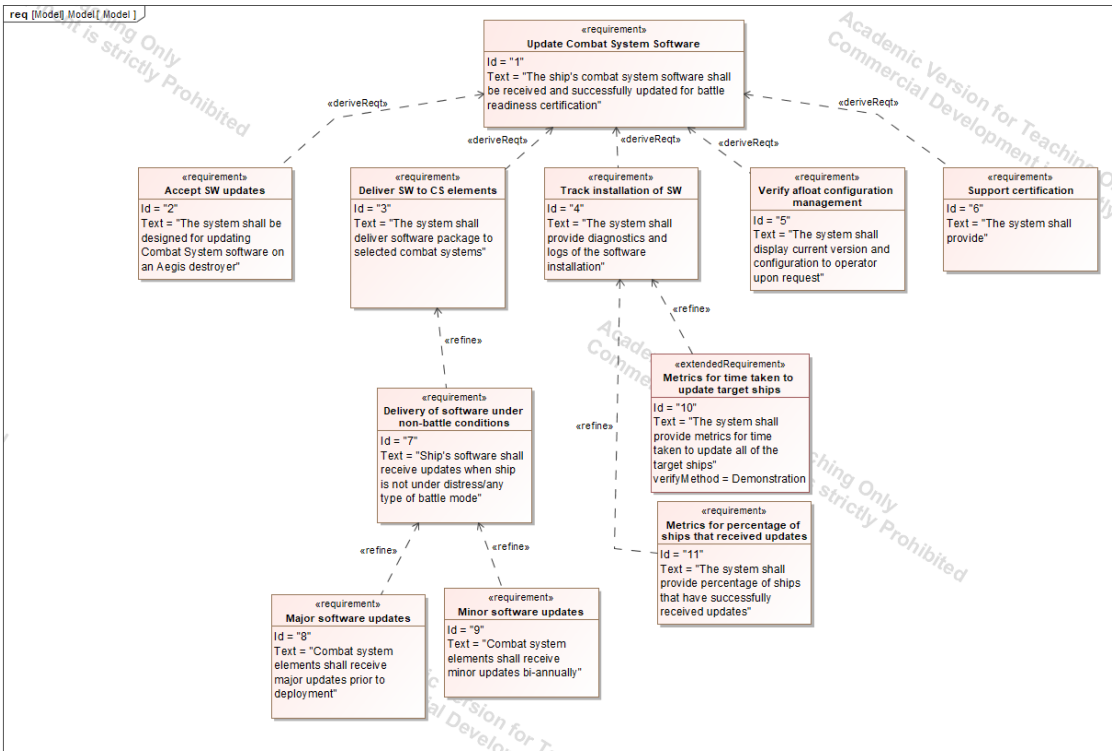

Figure 8. Functional Requirements in SysML

4. Non-functional Requirements

Non-functional requirements describe the “ilities” of the proposed system. These requirements define system attributes such as security, reliability, performance, maintainability, scalability, and usability and are shown in Figure 9.

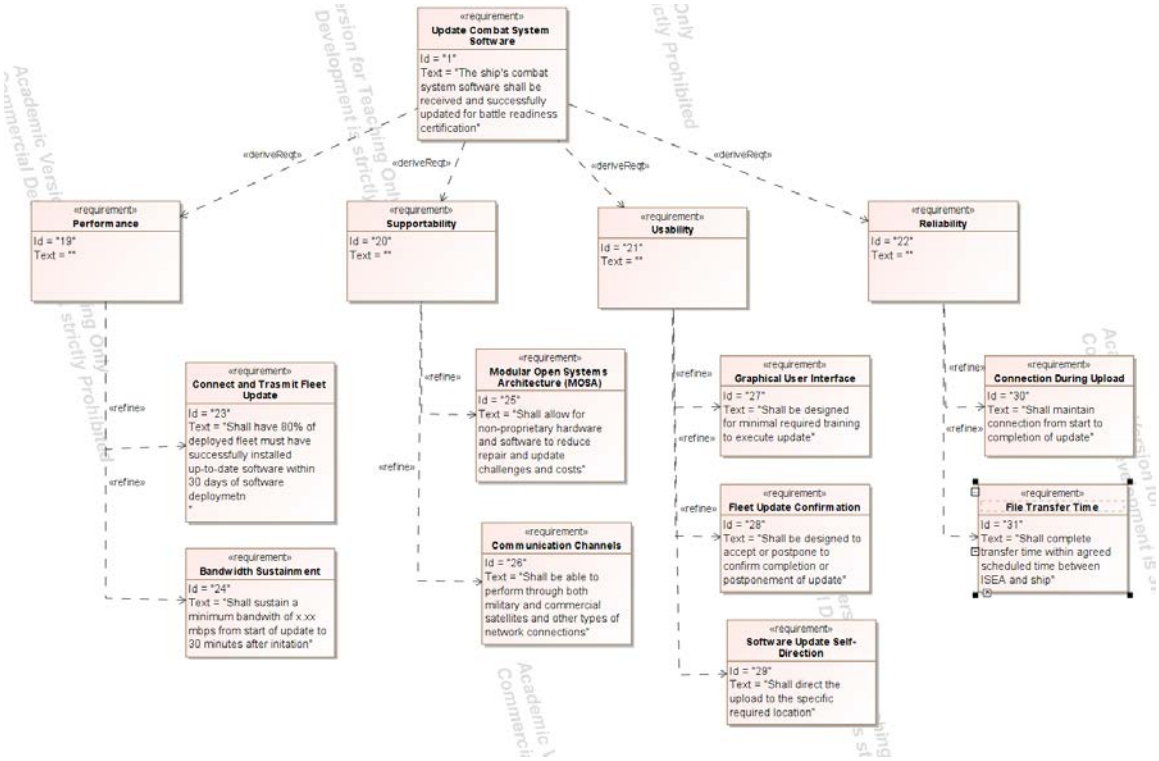

Figure 9. Non-functional Requirements in SysML

a. Performance

In Figure 10, the Performance Diagram for the RDS is shown. The vision owner has defined a threshold value of performance for the RDS that it be able to update 80% of the available fleet at the same point in time by sustaining a minimum bandwidth of 1.00 Megabits per second (Mbps) from the start of the update until 30 minutes after initiation. The RDS must be able to connect and transmit regardless of sea conditions, environment, and force conditions. In order to preserve satellite communication (SATCOM) bandwidth during critical situations, the Aegis destroyer CONOPs will determine whether the update

will be accepted or deferred. If the ship personnel defer the software update, the deferment will expire after 30 days, and the RDS will automatically re-engage the ship for acceptance or deferral again. If operational conditions allow, ship personnel also have the option to perform a manual download request before the 30 days deferral has expired.

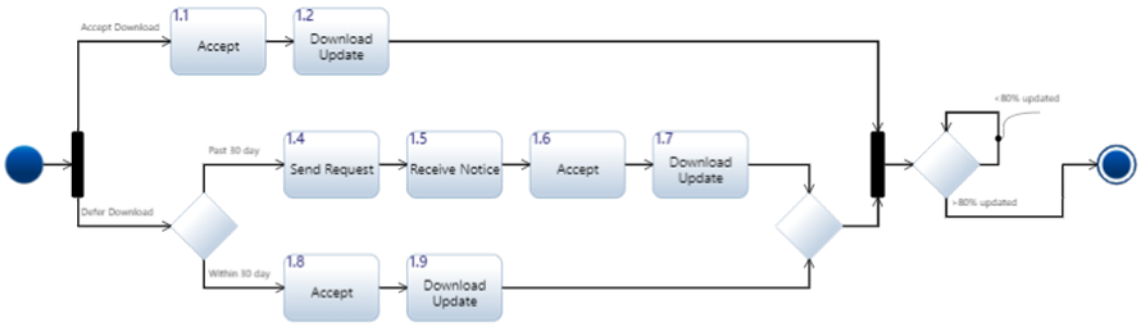

Figure 10. Performance Diagram

The system shall allow for interrupted connectivity when transmitting files to or from the ship. The system will resume transmission from point of disruption. In order to track fleet updates status, the system shall report the percentage of ships that have successfully executed the update(s). Additionally, the system will track the amount of time that it takes for the update to complete once initiated. The software update system shall contain code that directs the upload to the specific and required location, so that the user will not be required to perform any action other than what has been described in this subsection of the document. The software update system must be able to utilize and communicate with both military and commercial satellites and other types of network communication systems.

b. Supportability.

The RDS shall use COTS parts in order to simplify the supportability requirements during the system life cycle. The COTS design must be comprised of non-proprietary hardware and software. This construct will allow for trained technicians to perform repairs on ship-based system while deployed. Additionally, the COTS-based architecture will

allow for expansion of the remote software update system. Expansions will allow for additional CS to be integrated into the proposed system, thus reducing life cycle costs of the system and logistics burden of the U.S. Naval fleet. The performance requirement for the software update system to be able to utilize and communicate with both military and commercial satellites, and other types of network communication systems is to expand the usable amount of satellite systems for increased supportability to the system functionality (Blanchard 2011).

c. Usability

The graphical user interface (GUI) shall allow for minimal training to be required so that multiple people (land-based and ship-based) are able to perform basic actions required for the update to be performed. The upload of files to be transferred out must be designed and executed by certified technicians to preclude the possibility of improper file transfer. The update ship-based GUI shall be designed to be "Accept/Postpone" along with "OK" to confirm completion or postponement of update. This will be required in order to make update 'clear and simple' in the event that the fleet or vessel is in stressing environment so that user is certain of action and status (Cohen 2014).

d. Reliability

The system shall maintain connection from start to completion of update. If connection is broken, slowed below specified performance, or stopped, the update must recognize point of disruption and resume at that point. As an example, the update must be completed within three minutes for basic updates (below 15 Mb), seven minutes (between 15 and 20 Mb), and within ten minutes (>20 Mb). Larger file size transfers must be scheduled and only executed while connected through a SATCOM system to verify ability to transfer and accept. If transmission is broken during the prescribed time allotments, the transmission log will be referenced by the CS software and resume at that point (Cohen 2014).

e. Interface

The derived interface requirements specify that the system shall have:

1. Wireless data interface between the NSWC PHD shore facilities to a communication satellite.
2. Wireless data interface between the communication satellite and the Navy ships.
3. LAN interface from the ship's wireless receiver unit to a software management system/database.
4. LAN interface from software management system/database to the ship's CS.

f. Legal

All DOD regulations and rules will be adhered to in the design and implementation of the system and the transfer of the files. File transfer will not be performed on unverified and unsecure networks. All transfers must adhere to military regulations that specify rules and governances for electronic file transfer and uploads.

E. CHAPTER SUMMARY

This chapter described problem refinement to include needs analysis, the CONOPs, the gap analysis for both the current and proposed systems, and finally the requirements analysis. This analysis will be used for determining the best alternative for delivery of CS software. The alternatives analysis can be linked to the requirements analysis for traceability and V&V.

V. SYSTEM ARCHITECTURE

This chapter includes the models and analysis of the system of interest. The models were used to analyze the various methods to deliver CS software to the ship. The SysML diagrams show the system architecture, structure and the functional flow. These diagrams trace the requirements to ensure the system meets the needs of the stakeholders.

A. METHODOLOGY

The methodology is based on the CONOPS diagram. The RDS is modeled with the remote station, servers, ships, and the communication link between them. The most efficient way to model the system would be to match the components on the ships. Since ships will not be getting a hardware change, the system in design must be made to accommodate the current technology on the ships.

The model starts with the ship block or components on the ship, and this can include communication equipment, storage servers, main computer and CS computer. The ship's main computer will receive an alert notification from the remote station server informing that a CS update is ready for download and install. The remote station will receive status updates regarding the download and installation of the software. The final notification will be for a successful installation.

The remote station will consist of the main computer where the software will get uploaded to the servers and become available for the ships to download. The remote station will consist of communication equipment that will allow it to connect to the ship's network. The remote station will also consist of another set of servers that will record all the software version configuration of the ships.

1. System Models

The system modeling methodology followed the modeling taxonomy adopted from the *Practical Guide to SysML* by Friedenthal, Moore, and Steiner (2015). The SysML diagrams follow the structure and behavior diagrams as identified in Figure 11. The models and diagrams are created in Cameo Enterprise Architecture software. For this project, the

major diagrams for the SysML model used are the structure diagrams, the behavior diagrams, and the requirements diagram. The structure of the RDS is defined in BDDs and IBDs. The structure diagrams show the components of the systems or subsystems and the interfaces between them to fulfil requirements. The behavior diagrams show describes the means of how the system will interface with other systems or subsystems. For this project, the behavior diagrams will describe how requests are transmitted, how software data is moved, and how confirmation receipts are received. These diagrams are traced back to the requirements diagram to ensure that all requirements are satisfied and verifiable.

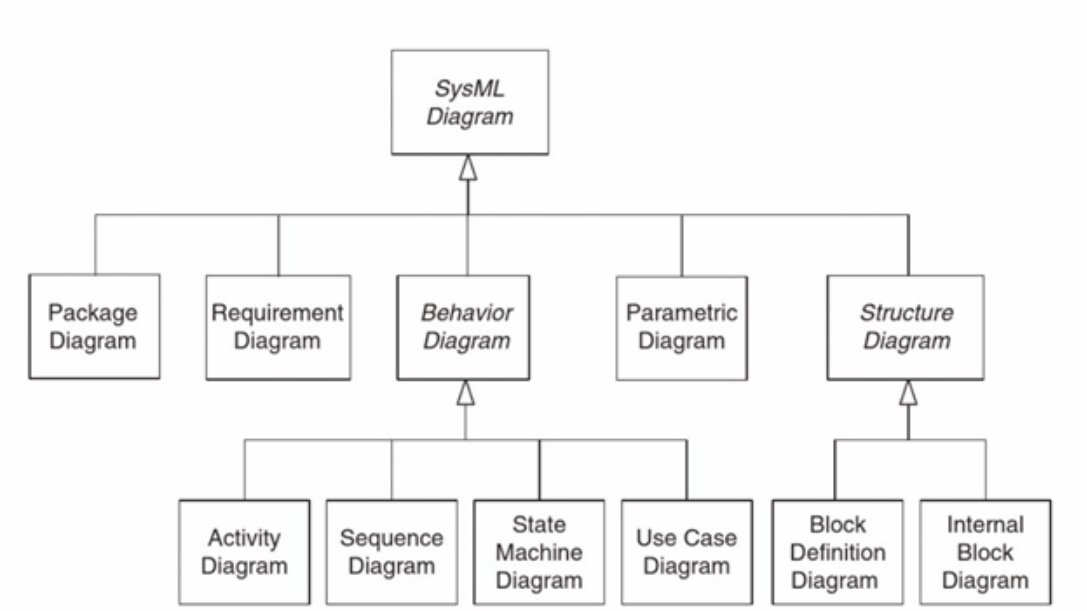

Figure 11. RDS System Models Including Behavior and Structure Diagrams

2. State Machine Diagram

The state machine diagram shown in Figure 12 displays the various actions that take place as part of the software update function and it is followed up with the software version check function. The actions of the left are for the software update function and the actions on the right are for software version check. If a software update is required, the state machine follows the left path. Upon successful installation of the software, the state transitions to the software version check. If a software update is not required but there is a

request to check the current software version installed on the ship, the path on the right would be chosen where the software version is verified without installation of new data.

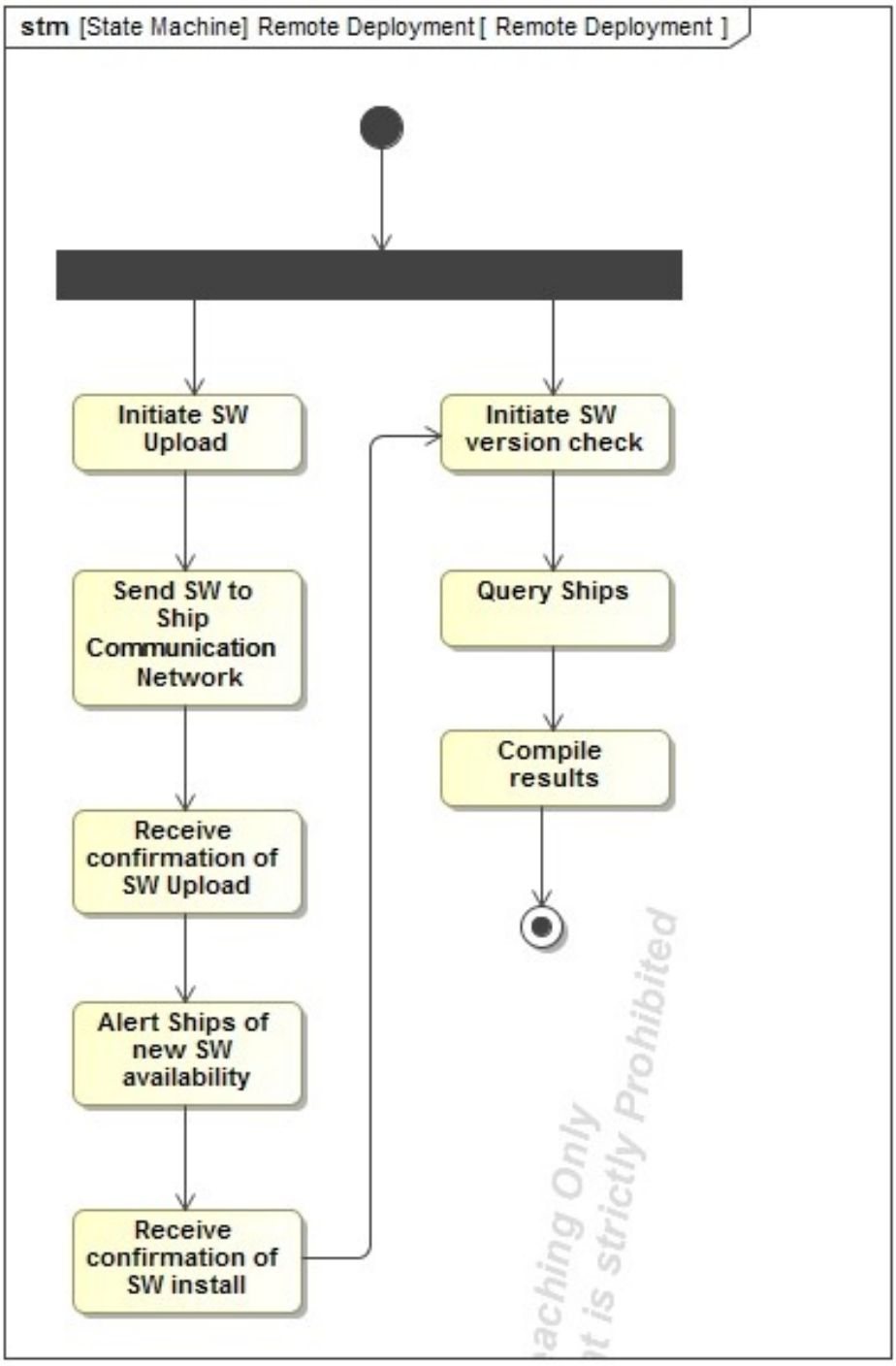

Figure 12. State Machine Diagram for the Software Update Process

3. System Boundary Model

The system boundary model shown in Figure 13 shows the boundary of the deployment system. External to the deployment system is the software generation and the CS certification. The software generators are involved in compiling and delivering the software to remote ground station where it enters the boundary of our system. The system exits the boundary into CS certification where the software is validated and verified by the RMF accreditation team.

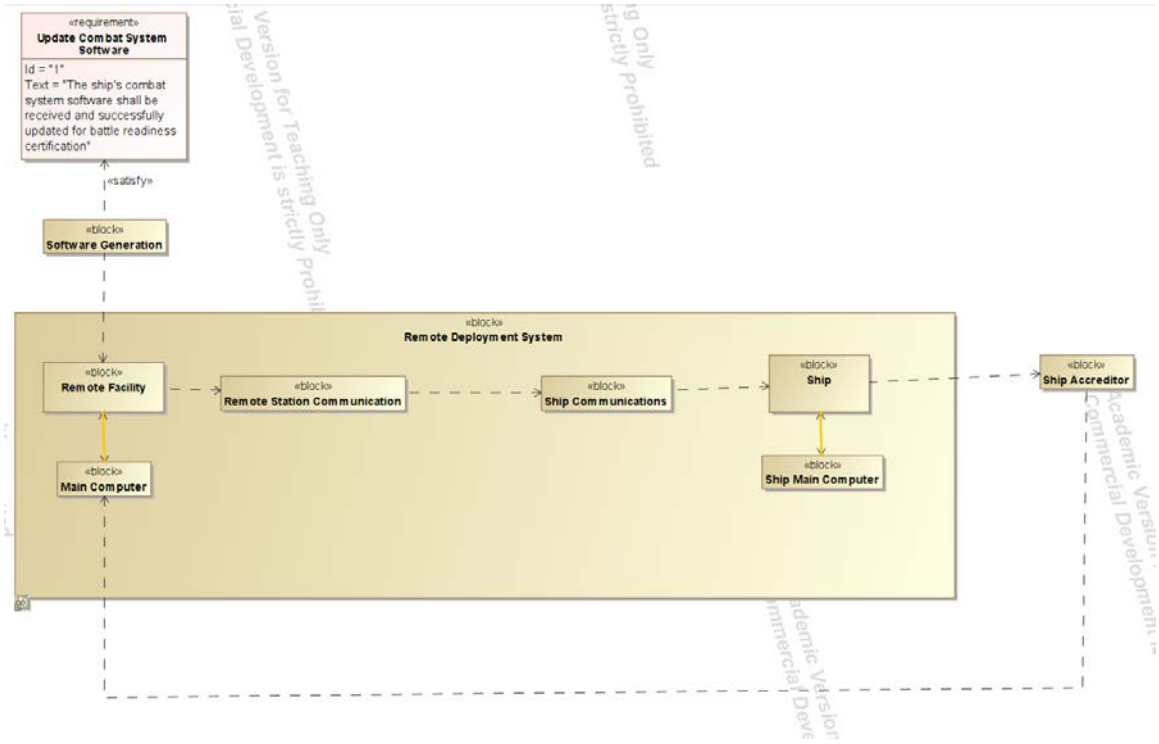

Figure 13. System Boundary Model

4. Use Case

The UC diagram shown in Figure 14 shows how three different groups of users will interact with the RDS to perform the actions. The UC starts with NSWC PHD receiving new CS software from the software developers. The operator at the remote station will input the software into the main computer and initiate upload to the local servers first after a virus scan is conducted on the software. The local servers will receive and host the

software for all the ships. The user can also initiate a software version check for the installed software on a ship. Afterwards, the ship will receive a notification of new software availability. Ship personnel will initiate software installation and wait for it to finish. Once the installation is successful, a notification will be sent to NSWC PHD informing the personnel that the installation was successful. The user at NSWC PHD will relay this information to the RMF accreditor, who will begin the accreditation process and certify the ship to operate with the new software. NSWC PHD will be notified after the CS has been certified for use.

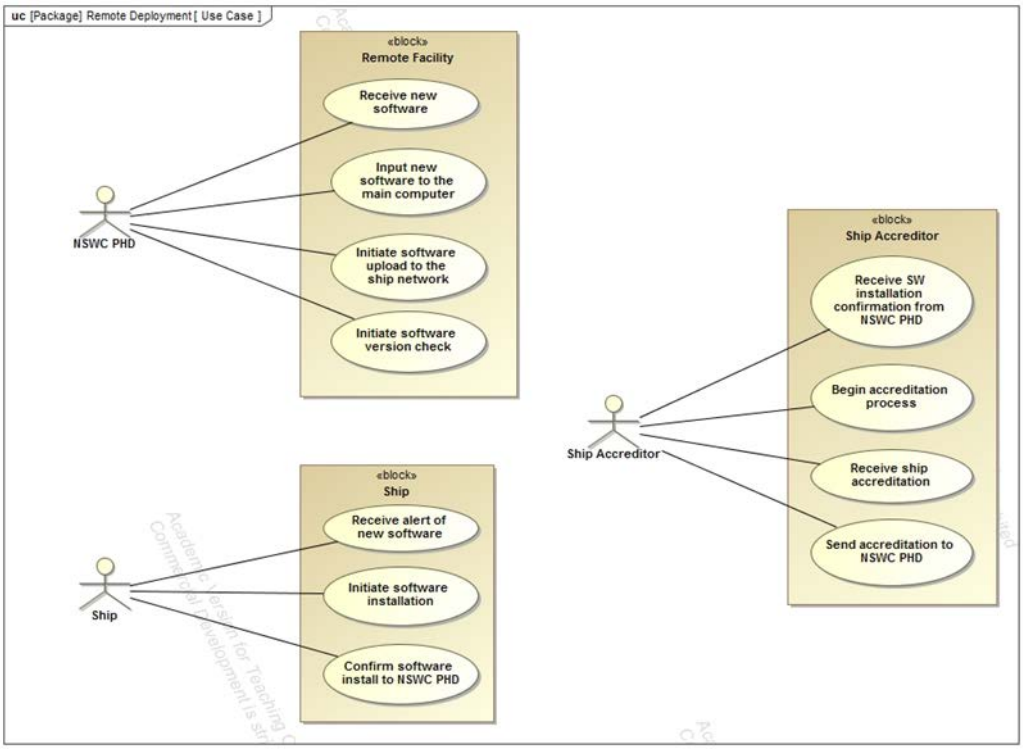

Figure 14. UC Diagram of the RDS

5. Action Diagrams

SysML diagrams were used to assist in building the architecture for this project. State machine and UC diagrams were shown in the earlier sections. In the AoA section, BDDs and IBDs are used to show the relevant blocks and the flow of information between them.

B. FUNCTIONAL ANALYSIS

The system will be capable of performing two main tasks as shown in Figure 15: software update and software version check. The software update function of the system will be conducted using the main computer interfacing with the ship's network and transferring files. Ship personnel will have the option to install the new software update as soon as it is available or later. The software version check function can be executed from the main computer of the system and will send a query through the communication network to each ship. The ship will return the query with the current software version information. This information will be provided to CM teams at NSWC PHD to ensure each ship has the correct software version installed. A sample of the CMP is shown in Appendix A. This plan will define the organizations involved, their roles in the system, and how to properly check and document current configuration onboard each ship.

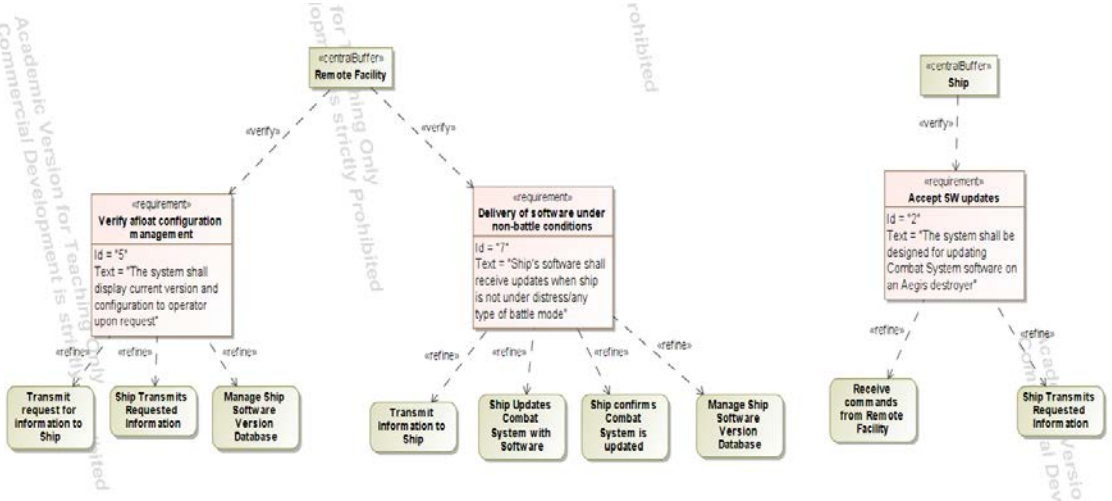

Figure 15. Functional Block Diagram Hierarchy

C. CHAPTER SUMMARY

This chapter described the modeling methodology used for the project's SysML models and the architecture. The chapter explained the boundaries of the system and what external systems interact with the RDS system. The state machine, UC, and functional analysis diagrams show how the system will operate. These models will be used for the AoA to determine the best solution that fits the need.

VI. ANALYSIS OF ALTERNATIVES

This chapter describes the different alternatives for deploying a software package to the ship. These solutions are different from the current process of hand carrying the package to the ship for installation. Each alternative will be scored based on the value model. For this analysis, the software will only be installed when a ship is in-port as it will affect the CS availability; therefore, it is crucial to conduct updates only when in-port.

A. POTENTIAL SOLUTIONS

1. Satellite (in port)

Figure 16 shows BDD of the RDS using SATCOM. The BDD expresses information about the system's structure (Delligatti 2014). The main computer in the remote station will receive software package from the software generators. Afterwards, the new software package will get pushed to the remote station communicating using the data transfer interface. The remote station communication will be used to host the new software for all the ships to download. SATCOM will be used to transfer the files to the ships. The ships will have an option to utilize the wired Ethernet cable while in-port to assist with the CS software updates as backup if available. The ship will manually confirm the download and allow installation. Upon successful installation, the ships will send the latest version number back to the main computer at NSWC PHD.

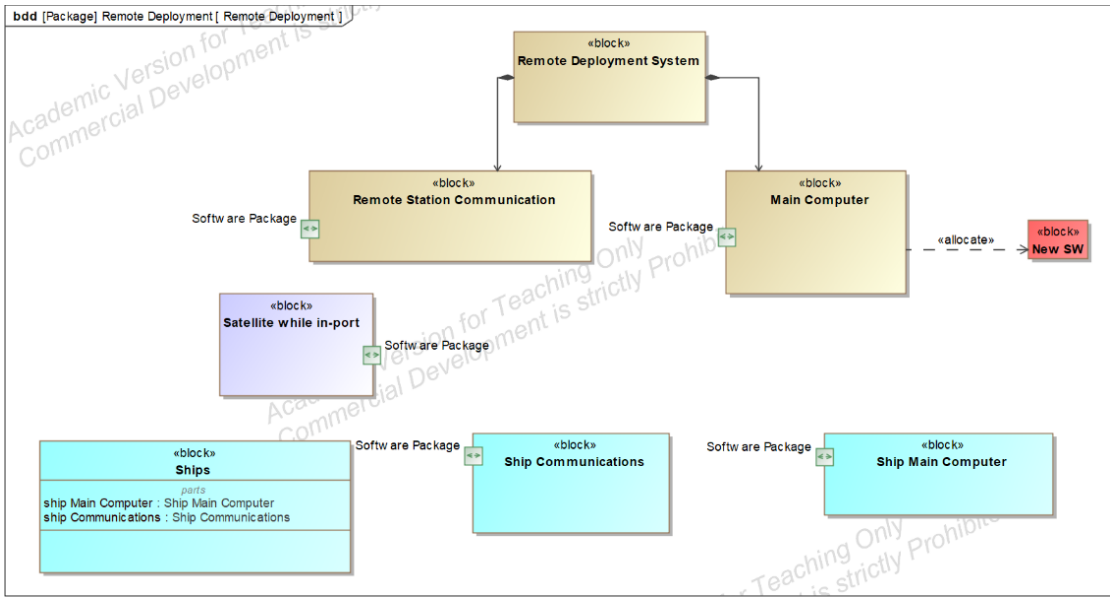

Figure 16. BDD for Satellite Alternative

Figure 17 shows the IBD for the Satellite Alternative. The update of the CS with the new software starts when a user manually inputs the software into the main computer using the CD drive. The main computer will read and upload the data to the remote station communication using data ports or LAN ports. The remote station communications will relay data to and from the internet using LAN ports as well. The internet block here serves as a pathway to connect various ships to the same network and also as a host of the files, in case there is a technical problem at the remote station. The ship's communication equipment will use satellites to download the software and move it to the ship's main computer for installation. Once installation is complete, the ship's main computer will send a notification back to the remote station indicating a successful installation. This IBD shows a two-way path between the remote station and the ships to ensure complete information can be relayed between the platforms.

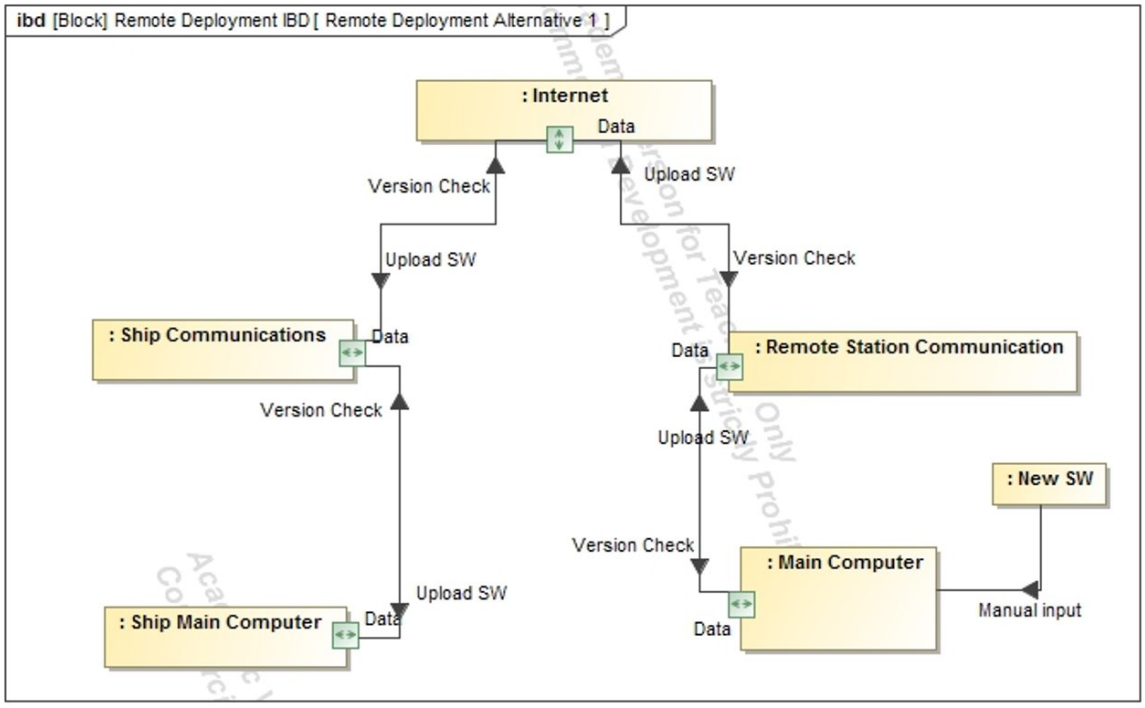

Figure 17. IBD for Satellite Alternative

2. Proximity Radio Frequency Link (in port)

Figure 18 shows the BDD for the proximity radio frequency (RF) link. This option will utilize line-of-sight radio to transfer data from the remote station to the nearby ships. The main computer in the remote station will receive a software package from the software generators. Afterwards, the new software package will get pushed to the remote station communicating using the data transfer interface. The remote station communication will interface with the proximity radio equipment. This requires line of sight for stable communication. The proximity radio equipment will be used to push files to all the ships that can access it. The ship will manually confirm the download and allow installation. Upon successful installation, the ships will send the latest version number back to the main computer at NSWC PHD.

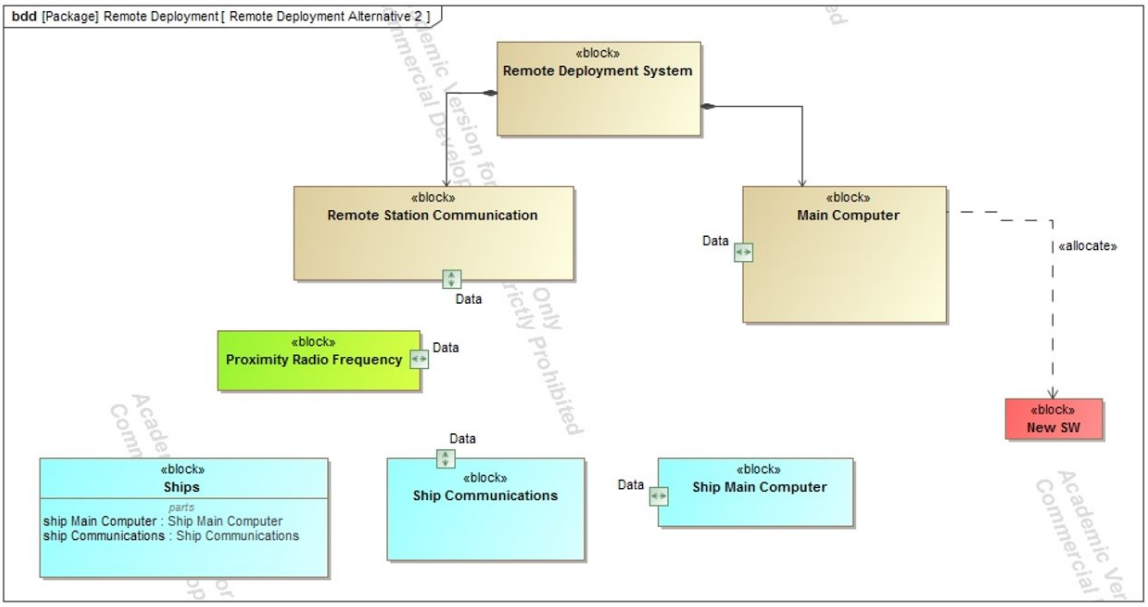

Figure 18. BDD for Proximity Radio Alternative

Figure 19 shows the IBD for the proximity RF alternative. The software installation starts when a user manually inputs the software into the main computer using the CD-drive. The main computer will read and upload the data to the Remote Station Communication using data ports or LAN ports. The remote station communications will relay data to and from the ships using RF transmitting and receiving equipment. The ship's communication equipment will interface with RF receivers and transmitters to download the software and move it to the ship's main computer for installation. Once installation is complete, the ship's main computer will send a notification back to the remote station indicating a successful installation. This IBD shows a two-way path between the remote station and the ships to ensure complete information can be relayed between the platforms.

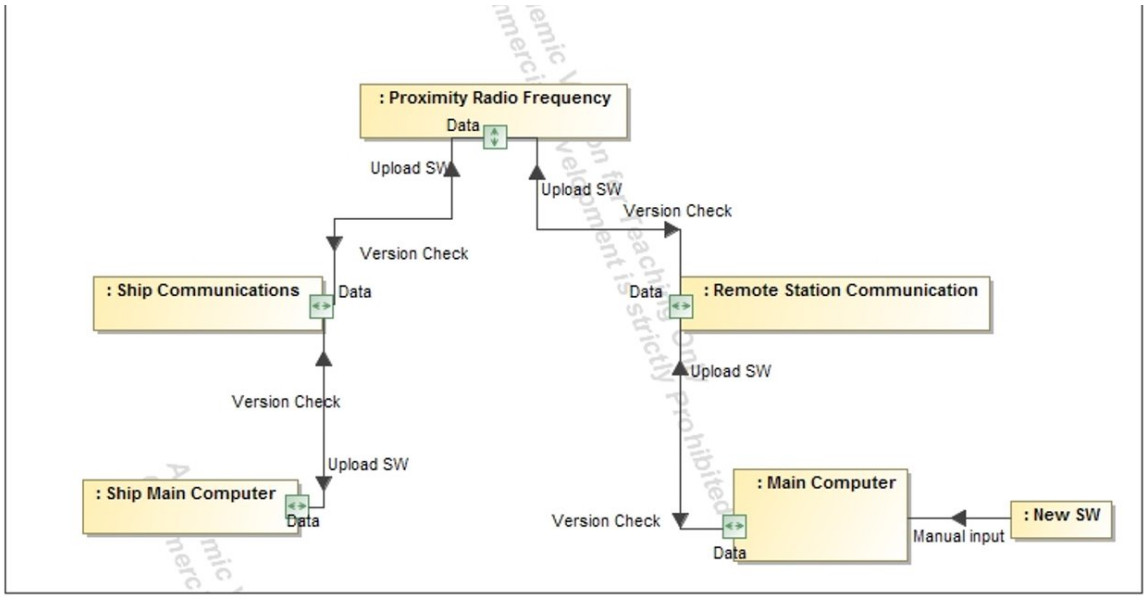

Figure 19. IBD for Radio Alternative

B. FEASIBILITY ANALYSIS

To determine the feasibility of using the RDS instead of the current method, a decision model was created. The decision model included a value model to help determine which alternative would be the best solution for NSWC PHD. The value model included a value hierarchy and expected relative system performance scores and weights in order to determine a total value score. The decision model is an additive value model and assumes the measures are preferentially independent. (Parnell, Driscoll, and Henderson 2011) Figure 20 shows the value hierarchy. The value hierarchy diagram shows the relationships between the effective need and evaluation measures for analysis. The effective need is to deliver software updates to the ships. The objectives are to maintain bandwidth, have a low number of operators to execute the software installation, and to have a minimal amount of time to setup and complete the installation.

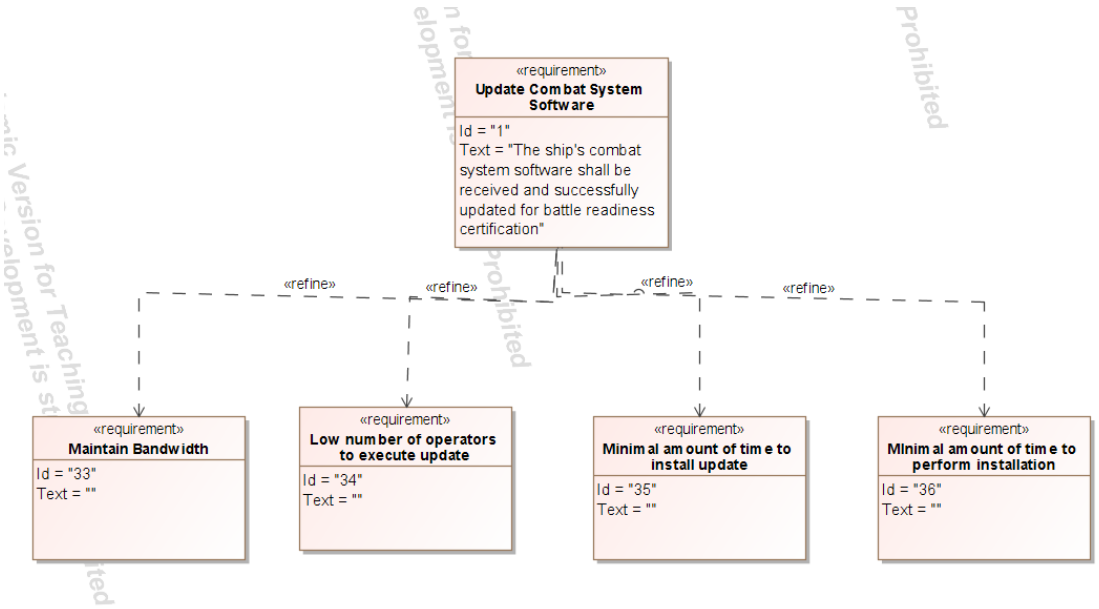

Figure 20. Value Hierarchy

The two alternatives are the current system of physically sending a team to the ship to install the software, and using a RDS. Three main objectives from the stakeholder are (1) minimize the number of personnel it takes to perform the software upgrade to the CS, (2) maximize the reliability of the system while the ship is in port, and (3) minimize the time needed to perform the upgrade. If there is a reduction in the number of personnel required, that means the cost of the installation will also go down. If the installation can be performed remotely, that saves on the number of people required and it saves on travel costs. The most important objective is the reliability of the method. If the method is not reliable, then it is not worthwhile to pursue. The current system is more reliable than the RDS because there are no issues with connectivity to the ship since a team physically with the upgrade will be onboard to not only perform the installation, but also troubleshoot any issues that may arise. Saving time is always a goal of the PO, especially if it means they can ensure the fleet is secure and upgraded in a timely manner before an issue arises. The RDS is a much faster process than the current system since there is no travel or coordination time with a team to go to the ship. Instead everything can be arranged virtually, and the ship can perform the installation when they have the most time, even if it is at night when most installation teams would not be onboard.

Table 1 shows a decision model matrix for alternative system selection. A relative expected system performance score was assigned to show how each objective is met by each alternative. The highest scoring alternative would have the most value to the command to implement. The criterion weights were based off of priorities expressed by the vision owner. The higher the priority, the higher the criterion weight. The "alternative selection data" was calculated differently depending on the category. For example, the "number of personnel required" has more value when fewer people are required. So the RDS has a higher value than the current system. For the reliability of the system, the more reliable it is, the more value it has to the command. Since the current system would not have to worry about interference or interruptions, it received a higher score than the RDS. However, as you will see though the modeling in the next section, the RDS was relatively reliable as well. The "sum scores of alternatives" was calculated by multiplying the criterion weight by the "alternative selection data" value. These values were then summed to provide a total value score for each alternative. The higher the score, the more value there is to the program.

Table 1. Decision Model for Alternative System Selection

		Alternative Selection Data		Sum Scores of Alternatives	
Evaluation method	Criterion Weight	Current System	RDS	Current System	RDS
# of personnel required	0.2	0.2	0.5	0.04	0.10
Reliability (Interference/Interruptions)	0.5	0.5	0.4	0.25	0.20
Time Required	0.3	0.2	0.6	0.06	0.18
				0.35	0.48

After performing the analysis with the criterion weights, it shows that the RDS is the preferred system. Now we need to determine which of the candidate RDS solutions to recommend through Monte Carlo simulations.

C. MONTE CARLO SIMULATION

1. Design of the Monte Carlo Simulation for the RDS

To determine expected system performance for the solutions alternatives, a Monte Carlo Simulation was created. Table 2 shows the different factors for the Monte Carlo analysis. The data transfer speed is dependent on the amount of bandwidth between the remote deployment station communication equipment, the communication system, and the ship's communication equipment. The software file size can vary to a small file to a relatively large file. Additionally, interference and interruptions will need to be handled by each alternative. The Monte Carlo simulation setup is described in Appendix C.

Table 2. Parameters Used in Monte Carlo Setup

Factor	Low	Medium	High
Transfer Speed	0.1 Mbps	5 Mbps	15 Mbps
Software File Size	1 MB	500 MB	1 GB
Interference/Interruptions	None	Some	Many

2. Simulation Results

The Monte Carlo analysis was conducted using the Innoslate software for measuring the average download times for the speeds shown in Table 2. An action diagram was created to map out the download process and the data speeds and file size were chosen at random. The results are shown in Figure 21 and Figure 22.

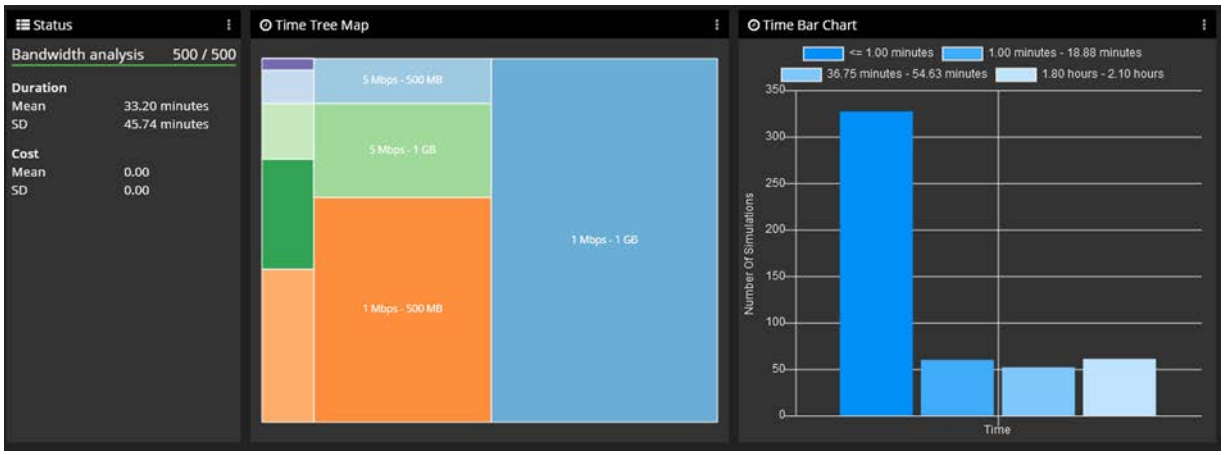

Figure 21. Monte Carlo Simulation Results for 500 Runs

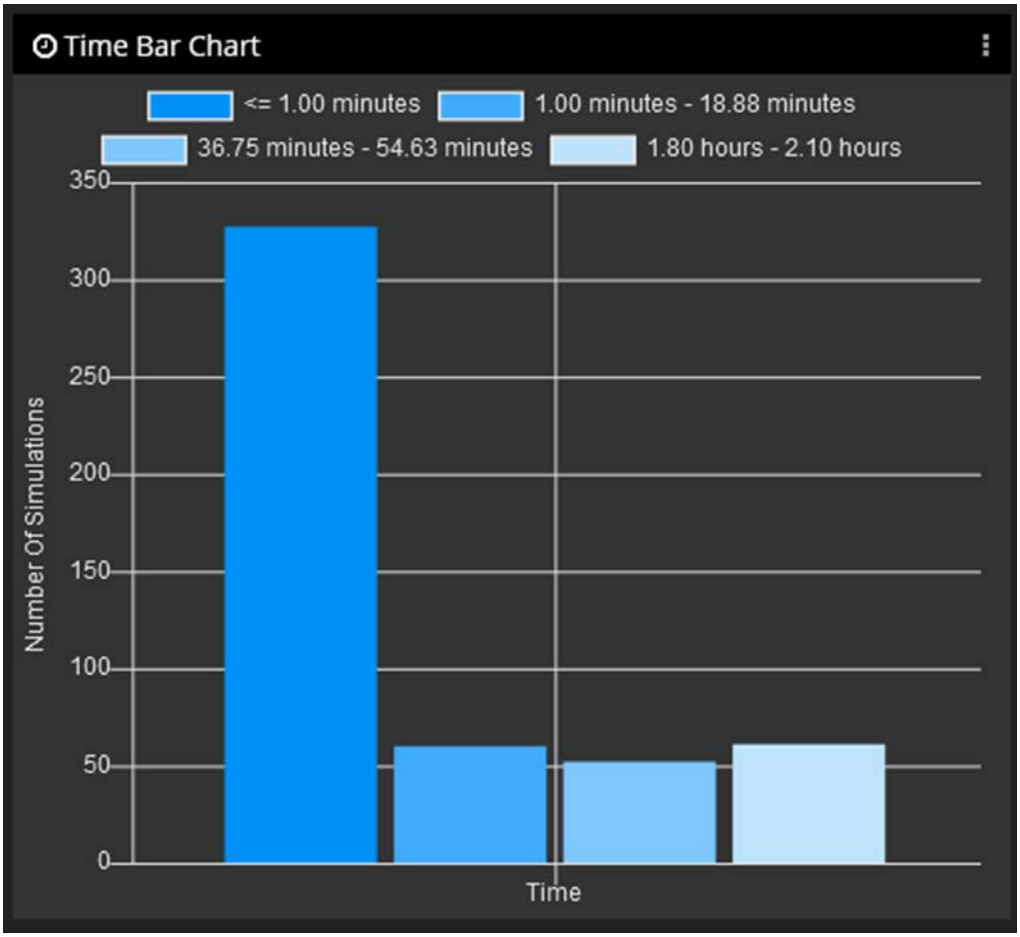

Figure 22. Monte Carlo Results Time Bar Chart

Figure 23 shows the numerical results. There were a total of 500 runs conducted and in 327 runs, the download time was less than a minute. There were 60 runs that took between one to 19 minutes, 52 runs that took between 37 to 55 minutes, and 61 runs that took around two hours. This data shows that majority of the download times will be less than two hours but there will be instances where it takes two hours to download. As expected, the one Mbps speed combined with the one gigabyte (GB) file was the slowest with around 2.4 hours. The one Mbps speed was chosen to represent a low bandwidth availability scenario, five Mbps for medium bandwidth, and 15 for high bandwidth availability scenario. This test helps us confirm our expectations and allows us to better design the system with large files being transferred over low bandwidth scenarios.

1 Mbps		5 Mbps		15 Mbps	
Size	Download Time	Size	Download Time	Size	Download Time
100 MB	14 mins	100 MB	3 mins	100 MB	1 min
500 MB	1.17 hrs	500 MB	14 mins	500 MB	5 mins
1 GB	2.4 hrs	1 GB	29 mins	1 GB	10 mins

Runs	Download Time
327	< 1 min
60	1 - 18.88 mins
52	36.75 - 54.63 mins
61	1.88 hrs - 2.1 hrs
500	Total Runs

Figure 23. Monte Carlo Results Download Times

D. CHAPTER SUMMARY

This chapter describes the different alternatives for the RDS system. The analysis scenario is while the ship is in-port in non-battle conditions. A feasibility analysis determined that the RDS would perform better than the current system. In order to recommend a conceptual solution, an AoA was conducted using Monte Carlo simulation in Innoslate. Cost analysis will need to be conducted outside of this paper for future work.

VII. RESULTS AND CONCLUSIONS

This chapter summarizes the major findings and results that support the project objectives. It also includes additional insights that emerged and recommendations for future work.

The project objective was to demonstrate a process that would support common CS incremental capability software updates that is faster than today's process. The project deliverables included a feasibility analysis, a CONOPs diagram for this process (which was shown in Figure 6), and a CMP. The CMP is described in Appendix A. Initial system risks were are identified and are described in Appendix B.

A. RESULTS

The first section of the results covers the RDS model and the process to efficiently deliver CS software and the process for determining CS software configuration and obtaining the evidence needed for certification. The second section covers the Monte Carlo simulation results based on the models showing the feasibility of using this system which is faster than today's methods.

1. RDS Model

The need for a more effective solution for delivering CS software updates to the ship began from the understanding of the constraints of the current system. The current system of physically delivering software encounters scheduling and resource limitations based on availability from both the software delivery team and the ship. The RDS modeling demonstrated a means for transmitting software packages in a more effective parallel process that reduces the number of personnel required for each software update. However, the RDS system will require relying on secure network communications between the remote facility and the ship.

2. Monte Carlo Simulation

From the Monte Carlo simulation, the time to transmit the software update from the remote facility to the ship has a mean of 33 minutes and the standard deviation of 46 minutes. From these results, it appears that the ship will download the software update from the remote site in a time period between less than a minute and an hour and 19 minutes 68% of the time. The longest download time based on the estimated parameters is about two hours. These parameters were set based on estimated software size and transfer speed. Under operational conditions, the download time may differ based on exact software package size and allowed bandwidth during transmission.

B. RECOMMENDATIONS

1. Recommendations Based on the Current Work

Based on the findings from the modeling and simulation, deploying CS software from remote locations would reduce the amount of time and resources used to schedule time with the ship and sending personnel to install CS updates. The RDS would provide a faster response time if software updates require an expedited delivery. Demonstrated through modeling and simulation, the RDS has the potential of sending CS packages without scheduling personnel to travel to install packages.

2. Recommendations for Future Work

Elements outside the scope of this paper include CS package file size, bandwidth of ship's network, and communication links between remote facility networks and ship networks, reducing total ownership cost, and addressing DISA concerns. Further analysis on these aspects will produce a more refined estimate of transmission time using the Monte Carlo method. Based on the literature review, the C2C24 program allows a means to connect to the ship's network. In addition to the models and simulations found in this paper, remote deployment programs can be explored further

APPENDIX A. CONFIGURATION MANAGEMENT PLAN

A. PURPOSE

The purpose of this appendix is to outline the CMP regarding the RDS. The CM process will establish and maintain the consistency of system performance despite inconsistencies of circumstances that may occur throughout the life cycle (software changes, parts obsolescence, etc.)

Key goals of CM include:

- Identification and documentation of hardware and software elements of the system
- Traceability of the elements to the design and requirements
- Tracking and verification of changes made to the system throughout the life cycle
- Auditing and tracking the configuration or the baseline of the fielded systems
- Assuring supportability throughout the life cycle of the system

To achieve these goals, the baseline and a database are usually established to facilitate traceability and control. The CM approach in this appendix is based on CM Guidance (MIL-HDBK-61) and CM Standard Implementation Guide (SAE-GEIA-HB-649). The DoDI 5000.02 “Operation of the Adaptive Acquisition Framework” also states that a CMP should be included in the Systems Engineering Plan (SEP).

B. OBJECTIVE

The objective of a CMP is to document the changes made and provide configuration identification, change control and configuration audits.

- The CMP plan will define:

- Baselining configuration items (CI)
- Method of controlling modification of CI
- Reporting and recording of modification of CI
- Ensuring the correctness of CI thru audits
- Methods of storage, handling and delivering of data pertaining to CI

C. APPROACH

Due to the nature of software updates, the methods of controlling, reporting, recording auditing and data storage will be largely done in electronics format. Use of database accessible via the network is encouraged.

D. ORGANIZATION

The NSWC PHD shore facility should be the main actor in the CMP implementation. NSWC PHD should also have direct control over the database for CM.

E. TRAINING

Training should be provided to both NSWC PHD and ship personnel, regarding the request of change, the reporting and recording of change, and the configuration audit procedures.

APPENDIX B. RISK ANALYSIS

This appendix contains the risk analysis for the RDS. The risk analysis will be limited to the technical aspect only, as cost information is not of the scope of this study and can be a future subject. The risk analysis approach is based on process set forth by system engineering textbooks (Blanchard and Fabrycky 2011) and several DOD risk management guides (Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics 2006) (Office of the Deputy Assistant Secretary of Defense for Systems Engineering 2017).

A. THE RISK MANAGEMENT PROCESS

Risk is inherent to any engineering and program effort. The purpose of this section is to objectively present the identification and analysis on the risks associated with the RDS. The risk management process will be based on methodology presented in classical system engineering textbooks, such as the "System Engineering and Analysis" by Blanchard and Fabrycky, and on guidelines set forth by the "2006 Risk Management Guide for DOD Acquisition."

Blanchard and Fabrycky describe the four basic categories of risk: technical risk, cost risk, schedule risk and programmatic risk. For the purpose of this capstone, the main focus will be on the technical risks.

Risk management is an iterative process that is accomplished throughout the system engineering process. Blanchard and Fabrycky describe this process as follows (Blanchard and Fabrycky 2011, 692):

- Risk Planning - Includes the development of a risk management plan or a given program.
- Risk Identification - Includes the screening of all requirements and to identify those are likely not to be met. This capstone report will be focus on the technical risk indentation portion of this process.
- Risk Assessment - Pertains to determining the probability of failure to meet a specified requirement and the possible outcome and consequences of not meeting the requirement. This pertains to both the probability of occurrence and the estimated magnitude of the risk.

- Risk analysis - is accomplished to determine the options in which the risk can be eliminated or minimized.
- Risk handling - includes the activities associated with the incorporation of changes and system modification of product or process.

The capstone will also utilize the Risk Management Process defined by the DOD Risk Management Guidelines. The Risk Management Process uses the five key activities shown in Figure 24.

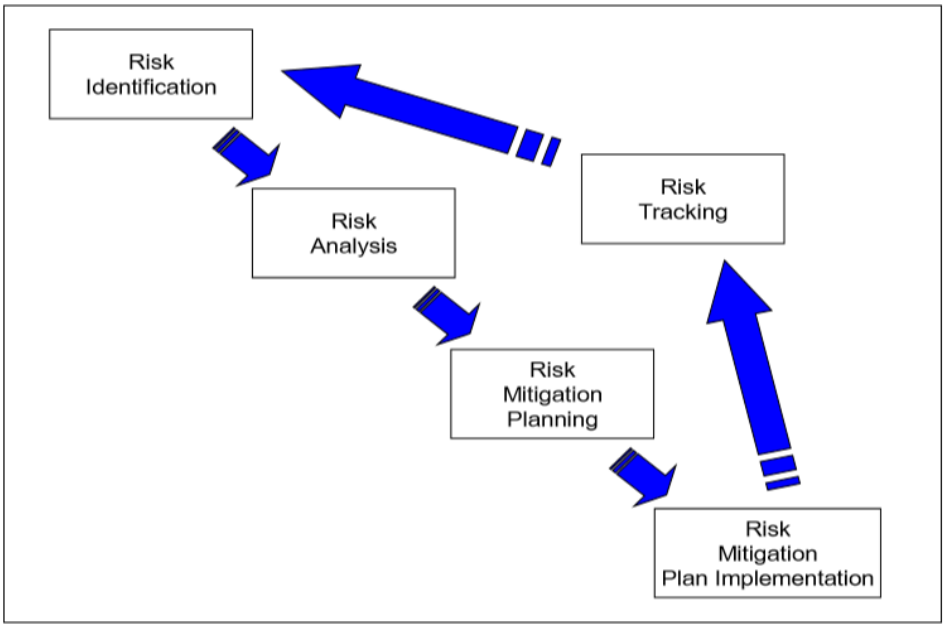

Figure 24. DOD Risk Management Process. Adapted from Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics (2006).

For the scope of this capstone, the risk management will be focus on the Risk Identification, Risk Analysis and Risk Mitigation Planning.

The 2006 *Risk Management Guide for DOD Acquisition* defines risk as "the measure of future uncertainties in achieving program performance goals and objectives within defined cost, schedule and performance constraints." The guide defines three components of risk as follows:

1. A root cause, which, if eliminated or corrected, would prevent a potential consequence from occurring
2. A probability (or likelihood) assessed at the present time of the root cause occurring
3. The consequence of that root cause occurrence

The probability of the root cause occurring can be quantified as numeric Risk Levels which facilitates the use of a risk matrix during risk analysis process. There are different ways of quantifying risk probability, but for this project, the team will use the 2006 *Risk Management Guide for DOD Acquisition*, which provides a five-level probability quantification.

Table 3. Levels of Risk Occurrence. Source: Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics (2006).

Level	Likelihood	Probability of Occurrence
1	Not Likely	~10%
2	Low Likelihood	~30%
3	Likely	~50%
4	Highly Likely	~70%
5	Near Certainty	~90%

Likelihood

The consequences are also quantified with five levels in the *DOD Risk Management Guide* with the five levels of consequence for technical performance shown in Table 4.

Table 4. Levels of Risk Consequences. Source: Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics (2006).

Consequence	Level	Technical Performance
	1	Minimal or no consequence to technical performance
	2	Minor reduction in technical performance or supportability, can be tolerated with little or no impact on program
	3	Moderate reduction in technical performance or supportability with limited impact on program objectives
	4	Significant degradation in technical performance or major shortfall in supportability; may jeopardize program success
	5	Severe degradation in technical performance; Cannot meet KPP or key technical/supportability threshold; will jeopardize program success

B. RISK IDENTIFICATION

1. Risk 1—Technical Maturity of RDS

The key technical aspects of the RDS, are the satellite network, network security, the data transmission format and associated error correction, and database manage and the integration of the whole system. All are implemented currently in the military and commercial sectors in some way. The associated probability of technical maturity issues is low. The consequences of such issue occurrence will not be negligible, but mitigation methods should be available either within the military or commercial industry. The level of risk occurrence should be low (1) and the consequence of risk is low to medium (2).

2. Risk 2—Stability of Connection and Correctness of Software Downloads

Several factors negatively impact the stability of connection of this system. Satellite connection is inherently less stable compared to land-based connection. A User Datagram Protocol (UDP) connection is inherently less stable than a Transmission Control protocol (TCP) connection, and ship is an unstable platform that can be operating in adverse weather conditions. The instability of connections not only increases the time needed to complete the download, but it may create associated software error. This issue is addressed by error correction mechanisms for UDP protocols. The frequency of occurrence for the stability and correctness issue will be high (5), but there are matured error correction methods mitigating the correctness issue, so the consequence of this risk should be medium (3) overall.

3. Risk 3—Network Security

Network security is a continuous and interactive process through the life cycle of network system due to the nature of evolving threats. Attempts on trying to breach military related networks never cease, but with the implementation of careful network security measures, the probably of a successful breach should be low (1). Due to the sensitive and classified nature of CS codes, if a breach happened, the consequences would be severe (5).

4. Risk 4—Training

Personnel training is always an integral part of any CS. The current process of CS software updates is performed by specialists trained by NSWC PHD, and CS software upgrades are a significant part of their daily workload, but under the structure of the RDS, this responsibility will be shifted to the ship's crew, and since CS upgrades are not going to be a daily task, adequate training of the associated procedures will be important. Fault-proof designs in the upgrade system (i.e., avoid sailor deleting the current CS without complete download and installation of the new CS software) should render the overall risk occurrence low to medium (2), and the associated consequences should be low to medium also (2).

5. Risk 5—Integration Issues

Remotely upgrading a CS will involve many sub-systems, with different manufacturers. Integration issues can potentially occur among the sub-systems. The risk for integration issues to occur should be medium (3), and the consequence of it should also be medium (3).

C. RISK ANALYSIS

With the above risk probably and consequences assignment, we can generate the risk matrix by mapping in the numeric values shown in Figure 25.

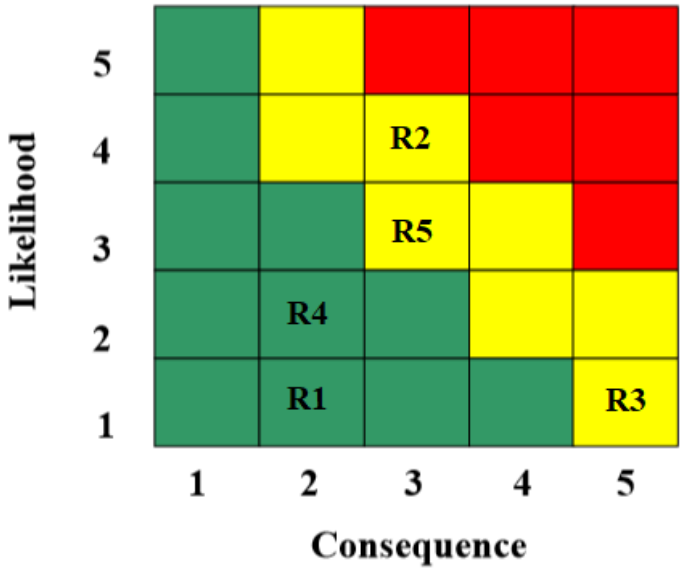

Figure 25. Risk Reporting Matrix

Risk 2, Stability of Connection and Correctness of Software Downloads, and Risk 3, Network Security, are the two risks that are closer to the high-risk area, and will need a well thought out RMP.

APPENDIX C. MONTE CARLO SIMULATION SETUP

The Monte Carlo simulation activity diagram shown in Figure 26 for the speed and file size analysis was conducted using Innoslate. The diagram was setup to randomly pick the file transfer speed first, then it would randomly pick the file size. The transfer speeds are one Mbps, five Mbps, and 15 Mbps. The file sizes are 100 Megabytes (MB), 500 MB, and one GB. The transfer time was determined by dividing the file size by the speed after bring converting the units for both into bits and bps using the binary method. The simulation would pick the speed first then the file size and show the number of times every combination of speed and size were selected.

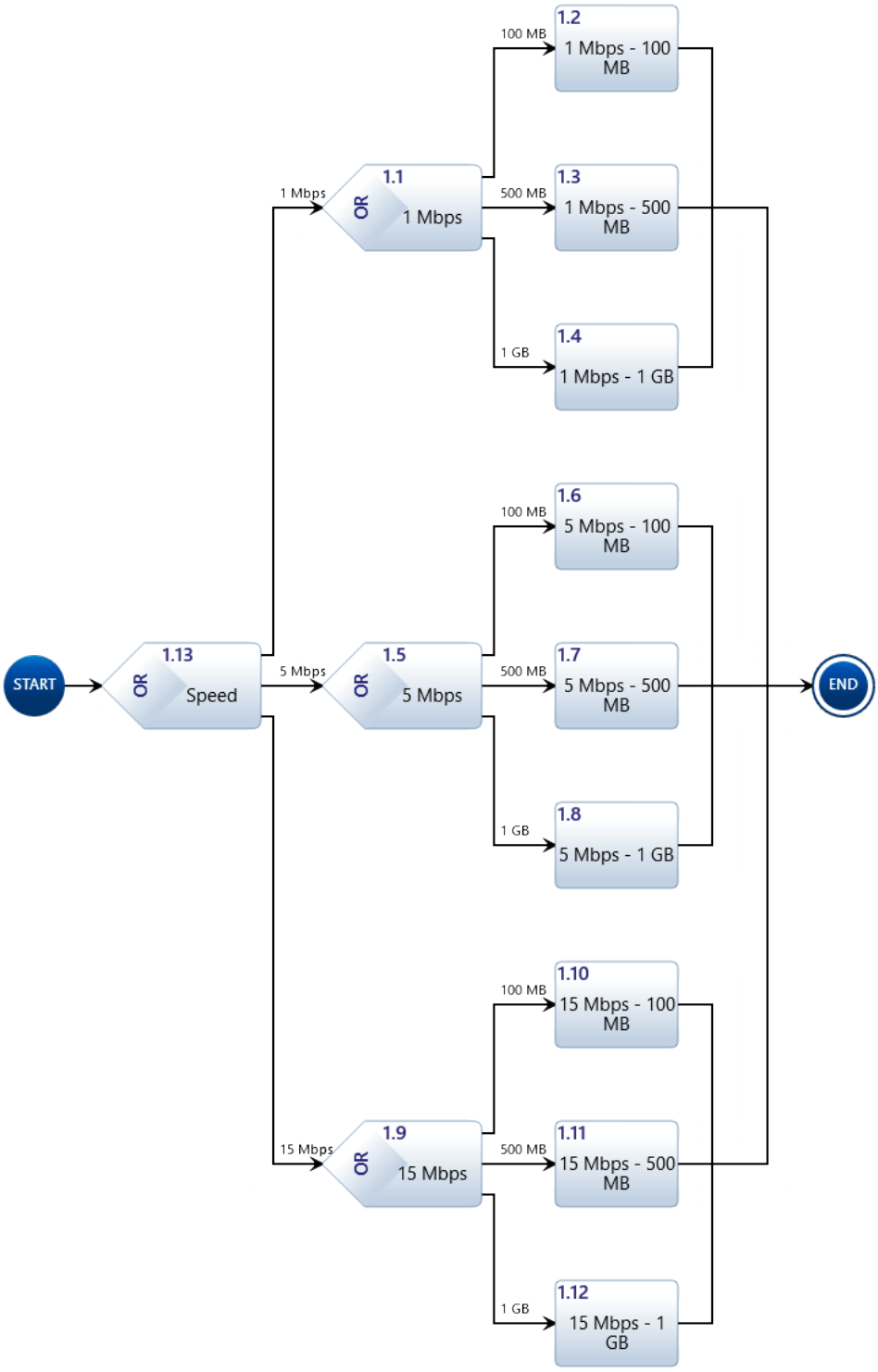

Figure 26. Monte Carlo Action Diagram

LIST OF REFERENCES

AcqNotes. 2018. *Information Technology: Risk Management Framework.* October 8. Accessed June 7, 2020. http://acqnotes.com/acqnote/careerfields/risk-management-framework-rmf-dod-information-technology.

Behrens, Colonel Anthony J. 2018. *Joint Air and Missile Defense Mission Command: A Singular Intelligent, Multi-Domain Platform and Culture.* Study, United States Naval War College.

Bendel, M. 2006. *An Introduction to Department of Defense IA Certification and Acreditation Process (DIACAP).* Study, Arlington: DOD.

Bender, Marc, Tom Maibaum, Mark Lawford, and Alan Wassyng. 2011. "Position Verification in the Context of Software/System Certification." *Proceedings of the 11th International Workshop on Automated Verification of Critical Systems.* 15.

Blanchard, Benjamin S, and Wolter J Fabrycky. 2011. *Systems Engineering and Analysis, 5th Edition.* Englewood Cliffs, NJ: Prentice Hall.

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

Cohen, Ryan, and Tao Wang. 2014. *GUI for Android Apps.* New York, New York: Heinz Weinheimer.

Delligatti, Lenny. 2014. *SysML Distilled: A Brief Guide to the Systems Modeling Language.* United Kingdom: Addison-Wesley.

Department of Defense. 2001. "MIL-HDBK-61A: Configuration Management Guidance." Washington, DC: Defense Printing Office, February 07.

———. 2012. "MIL-STD-882E Rev. 1 Department of Defense Standard Practice: System Safety." Washington, DC: Defense Printing Office, May 11.

Department of the Navy. 2018. "Navy Aims for "Compile to Combat in 24 Hours"." *CHIPS.*

Friendenthal, Sanford, Alan Moore, and Rick Steiner. 2015. *A Practical Guide to SysML: The Sytems Modeling Language.* Waltham, MA: Elsevier Inc.

Godwin, L., and J. Dewitt. 2005. "Redefining Tech Support." *Sea Power* 20–22.

Gonzales, Daniel; Hollywood, John; Kingston, Gina; Signori, David;. 2005. *Network-Centric Operations Case Study: Air-to-Air Combat With and Without Link 16.* monograph, Arlington: RAND Corporation, 106.

ISO/IEC/IEEE. 2018. “Standard 24748-1:2018.” *Systems and Software Engineering - Life Cycle Management - Part 1: Guidelenes for LIfe Cycle Management.* New York: International Organization for Standardization.

McCrady, Stuart G. 2013. *Designing SCADA Application Software: A Practical Approach.* New York: Elsevier.

McDermott, Kara. 2019. “SPAWAR Supports Navy’s Digital Transformation with ‘Compile to Combat in 24 Hours’ Training Series.” *CHIPS.*

Moore, Vice Adm Thomas J. 2019. *Technology Development Roadmap - Naval Power and Energy Systems.* Roadmap, NAVSEA, Arlington: U.S. Navy, 45.

Mugarza, Imanol, Jorge Parra, and Eduardo Jacob. 2018. “Analysis of Existing Dynamic Software Updating Techniques for Safe and Secure Industrial Control Systems.” *International Journal of Safety and Security Engineering* 121–131.

Office of the Deputy Assistant Secretary of Defense for Systems Engineering. 2017. “DOD Risk, Issue, and Opportunity Management Guide for Defense Acquisition Programs.” Washington, DC: Defense Printing Office, January.

Office of the Under Secretary of Defense for Acquisition and Sustainment. 2020. “DODI 5000.02 Operation of the Adaptive Acquisition Framework.” Washington, DC: Government Printing Office, January 23.

Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics. 2006. “Risk Management Guide for DOD Acquisition, Sixth Edition.” Washington, DC: Defense Printing Office, August.

Office of the Undersecretary of Defense for Systems Engineering. 2014. “Department of Defense Risk Management Guide for Defense Acquisition Programs.” Washington, DC: Defense Printing Office, December.

Parnell, Gregory S, Patrick J Driscoll, and Dale L Henderson. 2011. *Decision Making in Systems Engineering and Management.* Hoboken, NJ: John Wiley & Sons, Inc.

Rognlie, Aleron B. 2010. *An analysis of return on investment of the Consolidated Afloat Networks and Enterprise Services (CANES) program.* Thesis, Naval Postgraduate School, Monterey: Calhoun.

SEA06L. 2019. *Model-Based Product Support Overview.* Briefing, Arlington: NAVSEA, 23.

Society of Automotive Engineers. 2016. “GEIAHB649A: Configuration Management Standard Implementation Guide.” Warrendale, PA: SAE International, March 01.

Tesla. 2016. *Tesla Software Updates.* Hoefler and Company. Accessed 2020 29, 2020. https://www.tesla.com/support/software-updates.

Tonthat, H., J. Fung, J. Timm, and L. Godwin. 2005. “ORTSTARS: The Way Ahead for Distance Support.” *Naval Engineer Journal* 87–93.

Upadhyay, Darshana, and SrinivaS Sampalli. 2020. “SCADA (Supervisory Control and Data Acquisition) systems:.” *Computers and Security* 53–71.

Zhichao, Dai, and Ying, Xiang. 2010. “Design of Remote Upgrade of Equipment Monitoring System Software.” *Second International Conference on Information Technology and Computer Science.* IEEE. 462–465.

Zimmerman, Captain John D. 2016. “Lessons in Innovation: The SSBM Tactical Control System Upgrade.” *United States Naval Institute, Proceedings*, March: 85–87.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California